NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

**DENOISING OF ACOUSTIC SIGNALS
USING
WAVELET/WIENER BASED TECHNIQUES**

by

Coskun Cebeci

June 1998

Thesis Advisor:	Monique P. Fargues
Co-Advisor:	Ralph D. Hippenstiel

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1998

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
DENOISING OF ACOUSTIC SIGNALS USING WAVELET/WIENER BASED TECHNIQUES

5. FUNDING NUMBERS

6. AUTHOR(S) Cebeci, Coskun

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis investigates the use of combined Wavelet decomposition and Wiener filtering for the removal of noise from underwater acoustic signals. Several Wavelet/Wiener based denoising techniques are presented and their performances compared. Performances of the denoising algorithms are compared to those of Wiener filter and wavelet thresholding implementation and demonstrate that Wavelet/Wiener based methods are also a viable tool for the denoising of acoustic data under more restrictive conditions.

14. SUBJECT TERMS Acoustic Signals, Wavelets, Wiener Filter, Denoising, Aliasing.

15. NUMBER OF PAGES
76

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

DENOISING OF ACOUSTIC SIGNALS USING WAVELET/WIENER BASED TECHNIQUES

Coskun Cebeci
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1998

Author: Coskun Cebeci
Coskun Cebeci

Approved by:
Monique P. Fargues, Thesis Advisor

Ralph D. Hippenstiel
Ralph D. Hippenstiel, Co-Advisor

David C. Jenn
for Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis investigates the use of combined Wavelet decomposition and Wiener filtering for the removal of noise from underwater acoustic signals. Several Wavelet/Wiener based denoising techniques are presented and their performances compared. Performances of the denoising algorithms are compared to those of Wiener filter and wavelet thresholding implementation and demonstrate that Wavelet/Wiener based methods are also a viable tool for the denoising of acoustic data under more restrictive conditions.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my country and the Turkish Navy for giving me the opportunity to complete my graduate education.

I would like to thank Professor Monique P. Fargues and Professor Ralph D. Hippenstiel for their support and encouragement during the thesis research process.

Most importantly, I would like to thank my wife, Yesim, and my parents. Although many miles away, they always supported me with their love. And finally, I would like to dedicate this study to my sons, Omer Serdar and Erol Server, whose existence inspires and motivates me.

I. INTRODUCTION

Underwater acoustic signals are generally embedded in *ambient* and *self* noise. Ambient noise arises from all natural and man-made acoustic sources and is independent of the means used to measure it [1]. It exists in a particular region of the ocean and accounts for shipping traffic, marine life, wave motion, wind and numerous other sources. The second type of noise, self-noise, is generated by the receiving system, which in general consists of a passive sonar receiver and its platform. Usually self-noise can be minimized but ambient noise can not be completely avoided. Since noise degrades the collected data, it is desired to remove the noise from the contaminated signal before further processing, such as detection, estimation or classification, is conducted.

Detection of signals in noise is a well-researched subject. A Wiener filter is the optimum linear filter for detecting the presence of a deterministic signal in noise and can be designed to estimate any signal, given that the user has the noise characteristics [2].

Wavelet analysis has received increasing interest over the past two decades and found wide-spread use in signal processing applications. This is due to its multiresolution capabilities that permit analysis of data at different frequencies with multiple levels of resolution. Furthermore, wavelets provide the flexibility of choosing a particular wavelet function for a specific application. A large variety of orthogonal basis functions can be selected for the transformation of a signal. Results have shown that denoising is often much easier in the wavelet domain than in the time domain or other traditional transform domains. In fact, wavelet noise removal has been shown to be superior to techniques that remove noise by conventional filtering. Conventional or convolutional filtering broadens edges and hence it makes it difficult to retain sharp features of the original signal [3].

The primary goal of this thesis is to provide a new approach to the problem of removing additive noise from a given signal. In particular, this work will investigate the performance of a combined wavelet and Wiener filtering technique to denoise underwater acoustic signals.

This thesis consists of five chapters. Chapter II discusses the theory of wavelet analysis by analogy with the more familiar Fourier analysis. Chapter III gives a brief introduction to Wiener filtering and wavelet denoising. Chapter IV explains the algorithms and provides simulation results using synthetic data. Chapter V provides the summary, conclusions and proposes some extensions.

II. WAVELETS

Time-frequency signal representations map a one-dimensional time signal into a two-dimensional function of time and frequency. Thus, they display time domain and frequency domain information. This combination yields a more revealing temporal localization picture of the spectral components of a signal.

The work in this thesis is mainly based on wavelets, which is a very young and robust time-scale processing tool. Traditionally, Fourier analysis has been used as a robust processing tool in many signal processing applications. Thus, this chapter will first discuss Fourier analysis and then present wavelet analysis.

A. FOURIER ANALYSIS

Wavelet analysis can be viewed as an extension of Fourier analysis. Thus, this section first reviews the main definitions and properties of the Fourier and short-time Fourier transforms.

1. Fourier Series

Let *x(t)* be a periodic signal with a fundamental period T_0 (i.e., $f_0=1/T_0$ is the fundamental frequency). Then *x(t)* can be represented as an infinite sum of periodic complex exponential functions. The Fourier series representation of *x(t)* is given by:

$$x(t) = \sum_{n=-\infty}^{\infty} a_n e^{j2\pi n f_0 t} , \qquad (2.1)$$

$$a_n = \frac{1}{T_0} \int_{T_0} x(t) e^{-j2\pi n f_0 t} dt ; \qquad n = 0, \pm 1, \pm 2, \ldots \quad . \qquad (2.2)$$

Equations (2.1) and (2.2) are called as the *synthesis* and *analysis* equation, respectively. The Fourier series coefficient a_n provides the frequency domain information of the signal at the n^{th} harmonic of f_0. The coefficient a_0 is the average value of *x(t)* over one complete period and is given by:

$$a_0 = \frac{1}{T_0}\int_{T_0} x(t)dt, \qquad (2.3)$$

which is often referred to as the *dc* term of *x(t)*.

2. Fourier Transform

The Fourier transform of a continuous aperiodic signal is defined by:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt, \qquad (2.4)$$

which is referred to as the *analysis* equation. The inverse Fourier transform is given by:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df, \qquad (2.5)$$

which is the Fourier *synthesis* equation. Equations (2.4) and (2.5) are known as the Fourier transform pair. *X(f)* gives the decomposition of *x(t)* in terms of complex exponential functions of different frequencies.

3. Short-Time Fourier Transform

As can be seen from Equation (2.4), the analysis coefficients *X(f)* are computed as inner products of *x(t)* with sinusoidal basis functions of infinite duration. The result is that the Fourier transform is a quite robust method for the time-frequency analysis of narrow band signals (e.g., sinewaves). It is not matched to non-stationary signals, that is, signals whose frequency structures evolve in time. It lacks time localization of frequencies, which is especially very important for the analysis of signals of short duration. For non-stationary signals, it is necessary to obtain an accurate time-frequency representation of the signal that is a function of both time and frequency. By contrast the Fourier transform is a function of frequency only.

The short-time Fourier transform (STFT) was first proposed by Gabor [4]. In this transform an analysis window is chosen and this window divides the signal into short time segments where the portion of the signals are assumed to be stationary within the segments. Initially, the window is located at the beginning of the data allowing the computation of the Fourier transform for the first segment. Then, the window is shifted in time to process other segments of the data.

The STFT representation of a signal *x(t)* is:

$$S(\tau, f) = \int_{-\infty}^{\infty} x(t) w^*(t-\tau) e^{-j2\pi ft} dt, \qquad (2.6)$$

where *w(t)* is the analysis window function. The *spectrogram* of *x(t)* which provides the measure of the signal energy in the time-frequency surface can be obtained from *S(τ,f)* by taking its squared magnitude. The time and frequency resolution of the time-frequency surface are determined by:

$$\Delta t^2 = \frac{\int_{-\infty}^{\infty} t^2 |w(t)|^2 dt}{\int_{-\infty}^{\infty} |w(t)|^2 dt}, \qquad (2.7)$$

and by:

$$\Delta f^2 = \frac{\int_{-\infty}^{\infty} f^2 |W(f)|^2 df}{\int_{-\infty}^{\infty} |W(f)|^2 df}, \qquad (2.8)$$

where *W(f)* is the Fourier transform of *w(t)*.

Equations (2.7) and (2.8) indicate that two impulses in time can be separated only if they are more than *Δt* apart, and two tonals must be more than *Δf* apart in frequency in order to be discriminated. The drawback of the STFT is related to the Heisenberg principle (i.e., *uncertainty* principle) which is a resolution problem [4] and states that there is a natural limit to the resolution, which is given by:

$$\Delta t \Delta f \geq \frac{1}{4\pi}. \qquad (2.9)$$

Using a window of infinite length results in a Fourier transform that provides perfect frequency resolution but no time information. Using a short window will give better time resolution but poorer frequency resolution. Since the same analysis window is used for all frequencies, the time-frequency resolution is fixed (i.e., the duration of all basis functions are exactly equal) and the STFT has a time-frequency plane with a uniform tiling [5], as shown in Figure 2.1 and Figure 2.2.

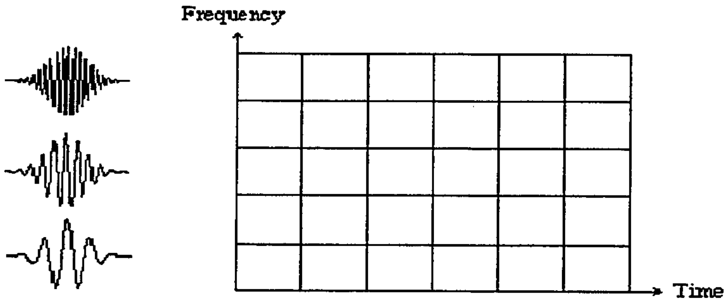


Figure 2.1: Uniform tiling of the time-frequency plane by the STFT with associated basis functions. After Ref. [5].

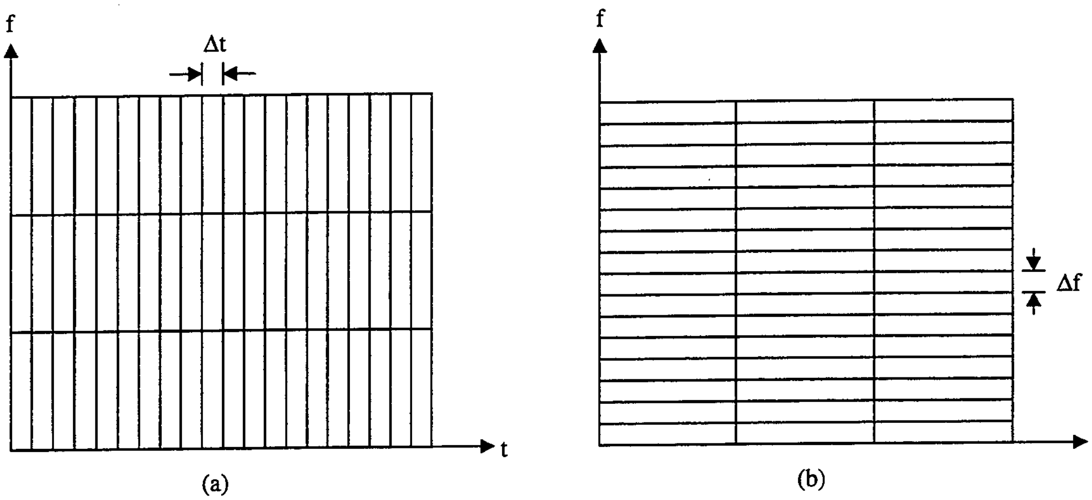


Figure 2.2: Time-Frequency plane for the STFT; (a) Narrow window, (b) Wide window.

Due to a *constant-bandwidth* filter operation, the time-frequency plane is partitioned into equal tiles [5]. The analysis filters have identical frequency responses [6]. Figure 2.3 shows the frequency response of the STFT.

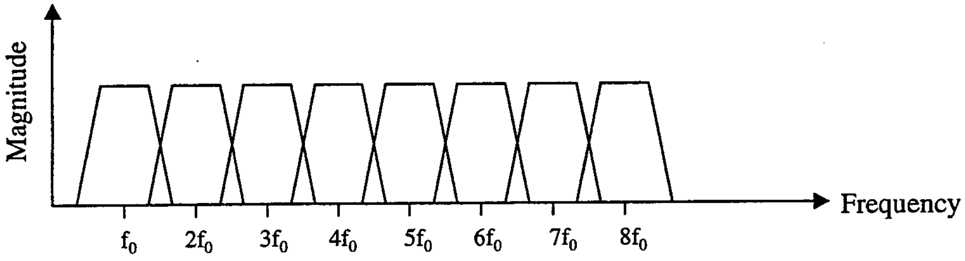


Figure 2.3: Frequency domain of the STFT. From Ref. [6].

When the window *w(t)* is Gaussian, the representation is called the Gabor transform [4]. The Gabor transform is optimal in terms of resolution since the Gaussian window has the best simultaneous time-frequency resolution for a given window length. This meets the lower bound with equality, that is:

$$\Delta t \Delta f = \frac{1}{4\pi}. \qquad (2.10)$$

The discrete Fourier transform (DFT) of a sequence *x(n)* takes the form:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \quad ; \quad k = 0,1,2,\ldots,N-1, \qquad (2.11)$$

and its inverse is as follows:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \quad ; \quad n = 0,1,2,\ldots,N-1. \qquad (2.12)$$

The discrete STFT is expressed as:

$$S(e^{j\omega}, m) = \sum_{n=-\infty}^{\infty} x(n) w(n-m) e^{-j\omega n}, \qquad (2.13)$$

where m is the center of the analysis window *w(n)*. The frequency variable ω is continuous, and takes the range $-\pi \le \omega \le \pi$.

B. WAVELET ANALYSIS

In the last decade, wavelet theory has become an important research area in many different fields of mathematics, physics and engineering. Advantages offered by wavelets include their efficient coding ability, their localization in time and frequency and their computational properties. Some results also claim a somewhat faster performance than that obtained by the Fast Fourier Transform. In this section, we will provide an introduction to wavelet analysis.

1. The Continuous Wavelet Transform

As described earlier, the STFT cannot provide a good resolution in time and frequency simultaneously. In other words, abrupt signal changes can only be detected within the chosen window length, as the time-frequency resolution is fixed.

If the signal of interest is composed of high frequency components of short duration plus low frequency components of long duration, then the multiresolution analysis, which analyses the signal at different frequencies with different resolution, is the proper tool. It provides good time but poor frequency resolution at high frequencies and good frequency but poor time resolution at low frequencies while the product $\Delta t\ \Delta f$ still satisfies the uncertainty principle [6].

The continuous wavelet transform (CWT) provides an alternative to the time-frequency information of the STFT as it decomposes a signal into a family of two parameter basis functions called *wavelets*. These functions include short duration-high frequency and long duration-low frequency components. Thus, time and frequency resolution variations are allowed although the time-frequency resolution product, given in Equation (2.10), remains constant. Each wavelet basis function is constructed from the same function, the so called *analyzing wavelet* or the *mother wavelet.*

The CWT of a signal *x(t)* with respect to the mother wavelet $\psi(t)$ is given by:

$$CWT(\tau,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^*(\frac{t-\tau}{a})dt\,, \tag{2.14}$$

where τ and a are translation and scaling parameters, respectively. The factor $1/\sqrt{a}$ is a normalization factor so that the energy of the scaled wavelet is the same as that of the mother wavelet. It can be seen that Equation (2.14) is an inner product form, which is commonly used to measure the correlation between the data segments and the basis functions.

The mother wavelet $\psi(t)$ must satisfy certain mathematical conditions. It must have compact support, be finite in duration, and have zero mean which requires that it must be oscillatory. The name *wavelet*, which means *little wave*, comes from its oscillation property [1,6]. The scaling factor a is directly related to the frequency resolution and inversely related to the time resolution of the CWT. Thus, as a increases (i.e., frequency decreases), the wavelet function becomes more stretched in the time domain and hence more compressed in the frequency domain. Similarly, as a decreases (i.e., frequency increases), $\psi(t)$ becomes more compressed in the time domain and thus more stretched in the frequency domain. This is shown in Figure 2.4. Due to this behavior, the CWT can be interpreted as a filter bank composed of band-pass filters with proportional bandwidth.

Thus, the terminology *time-scale plane* is preferred for the CWT, since the scaling factor a in wavelet analysis plays an analogous role to the inverse of the frequency in Fourier analysis.

The squared modulus of the CWT is a distribution of the signal energy in the time-scale plane and is called the *scalogram*. As illustrated in Figure 2.5, the scalogram has high time resolution at high frequency (low scale, small a) and high frequency resolution at low frequency (high scale, large a). The STFT has a constant time frequency tiling as seen in Figure 2.1.

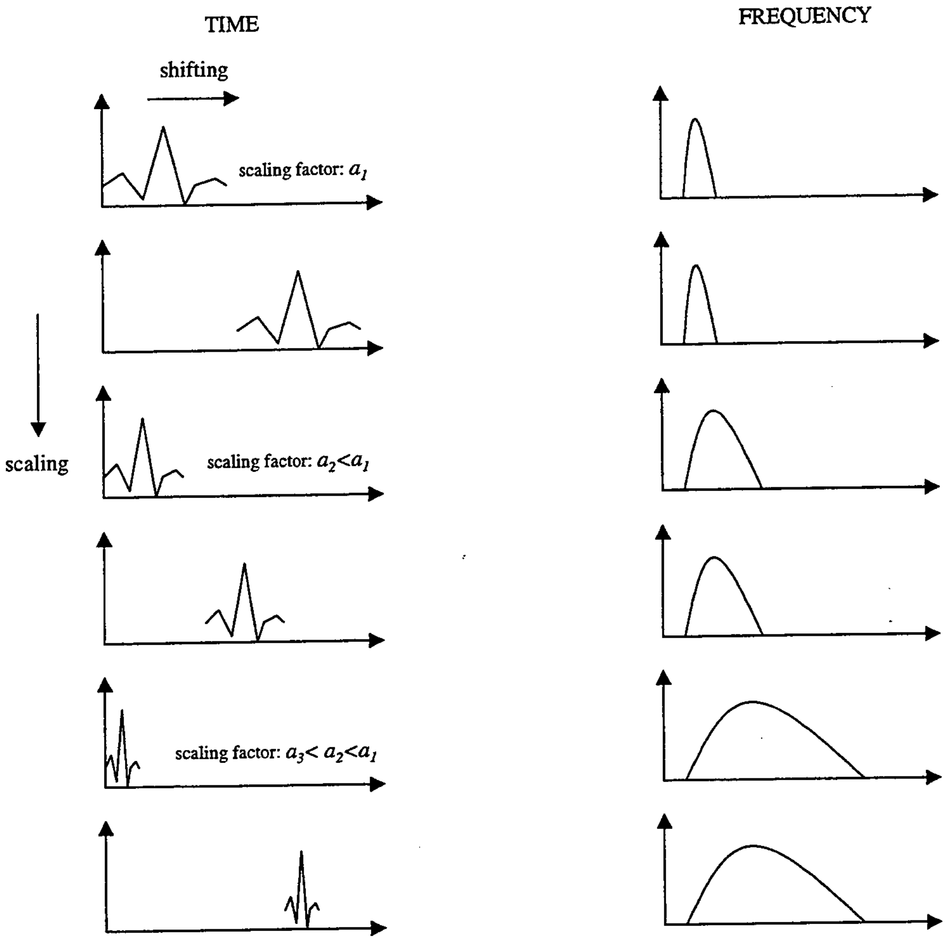


Figure 2.4: Scaling and shifting operations of wavelet transform in the time domain and corresponding magnitude behavior in the frequency domain.

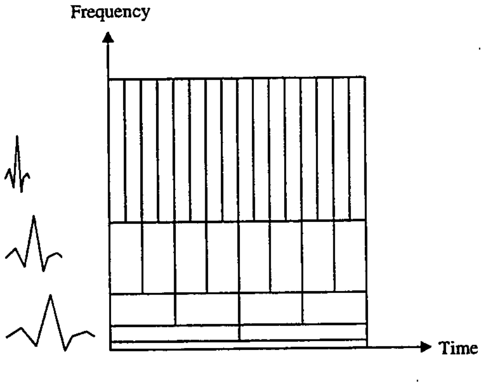


Figure 2.5: CWT Time-Frequency plane and associated basis functions. After Ref. [6].

A better understanding of high multiresolution capability of the scalogram compared to the spectrogram may be obtained from Figure 2.6. As can be seen from Figures 2.6a and 2.6c, the time localization for an impulse function at $t=t_0$ improves for low scales in the scalogram while it has a constant region due to the constant length of the analysis window in the spectrogram. For a signal composed of two sinusoids of frequencies f_0 and $2f_0$, the frequency resolution decreases with a in the scalogram as shown in Figure 2.6d.

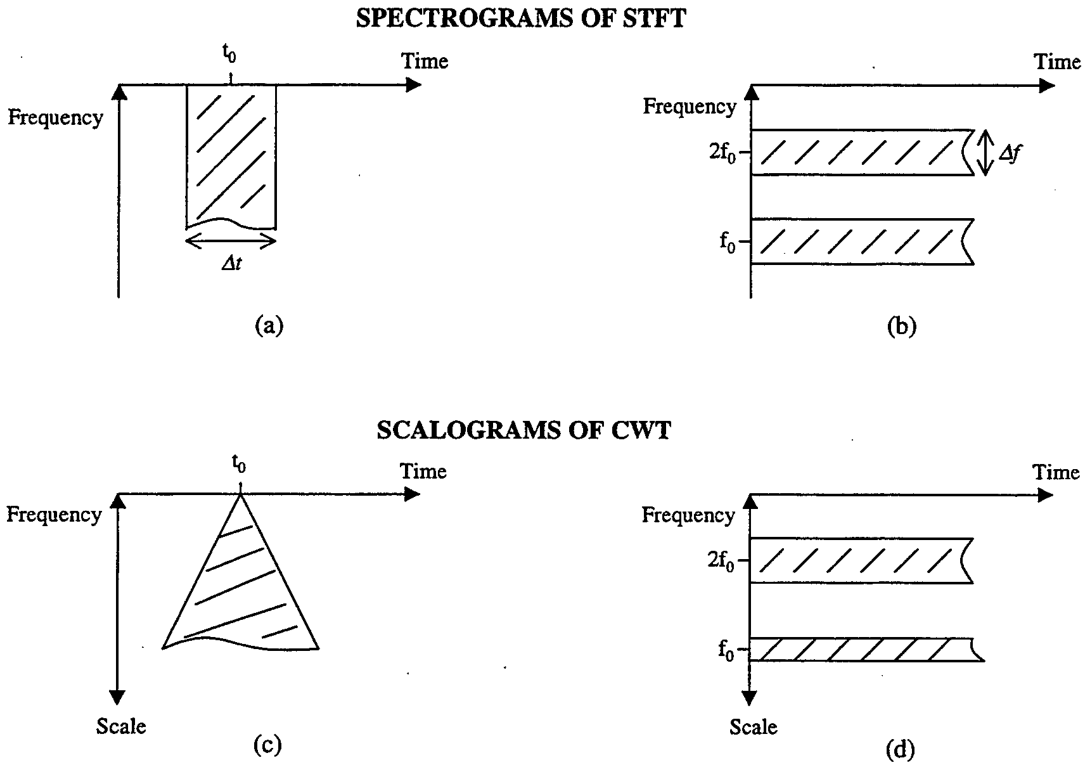


Figure 2.6: Comparison of spectrogram and scalogram for two signals. After Ref. [6].

2. The Discrete Wavelet Transform

The scaling and translation parameters in the CWT can take on any value. Thus, this kind of processing is redundant and computationally expensive due to the large amount of data involved. Furthermore, discretizing is a requirement to perform any transform on a computer for real-world applications. This section will provide a brief treatment of the discrete wavelet transform (DWT).

In general, when the energy of the signal is finite and an appropriate wavelet is used, not all values of the CWT decomposition are required to reconstruct the original signal exactly. Thus, a continuous-time signal can be represented by the knowledge of the discrete transform. Although continuous analysis is often easier to interpret, discrete analysis ensures space-saving coding and is sufficient for the synthesis [7].

The discrete form of the CWT is simply obtained by sampling the time-scale plane (i.e., by sampling τ and a) so that instead of choosing every possible scale and position, it uses only a subset of scales and positions at which calculations are made. Hence, for a discrete signal *x(n)* Equation (2.14) takes the form:

$$DWT(a,b) = \frac{1}{\sqrt{a}} \sum_{n} x(n) \psi^{*}(\frac{n-b}{a}), \qquad (2.15)$$

where *n, b* and *a* are the discrete versions of *t,* τ and *a*, respectively. The scaling factor *a* is discretized first at $a = a_0^J$; *J=0,1...* and the time parameter is discretized with respect to the scaling parameter at $b = ka = ka_0^J$ (*k* is an integer). Thus, the translation steps become proportional to the time scaling. This operation yields the decimated DWT at scale *J* and shift *k*.

Setting $a_0=2$ produces scales and positions, which are expressed as powers of two, and are called *dyadic* scales and positions. This is the most popular decomposition and is used in the standard DWT. This restriction makes the DWT analysis not only much more efficient but also provides stable reconstruction, very small reconstruction errors and fast algorithms [1]. Hence, Equation (2.15) can be rewritten as:

$$DWT(2^J, k2^J) = \frac{1}{\sqrt{2^J}} \sum_n x(n) \psi^*(2^{-J} n - k) = W_{J,k} . \tag{2.16}$$

An efficient way to implement this scheme using digital filtering techniques yields the so-called *fast wavelet transform* [7] and is known as Mallat's algorithm [8]. It uses a highpass filter (HPF) $h_1(n)$ and a lowpass filter (LPF) $h_0(n)$ which are also called *quadrature mirror filters* (QMF) where:

$$h_1(n) = (-1)^n h_0(P-1-n) , \tag{2.17}$$

$$|H_0(z)|^2 + |H_1(z)|^2 = 1, \tag{2.18}$$

where P is the order of the even sized filter. The filters are complementary filters which divide the frequency axis into two equal bands [0-$\pi/2$] and [$\pi/2$-π] radians, where π is half sampling frequency, as shown in Figure 2.7a. The signal is decomposed into different frequency bands by applying successive highpass and lowpass filtering operations. Then the filtered signal is decimated by two to remove the redundant information.

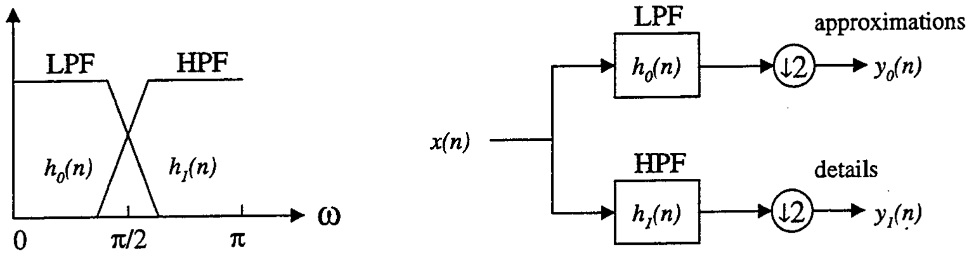
The output of the LPF branch is called the *approximation*, and contains the high scale, low frequency components of the signal given by:

$$y_0(k) = \sum_n x(n) h_0(2k-n) . \tag{2.19}$$

The output of the HPF branch is called the *detail* of the signal, and contains the low scale, high frequency components given by:

$$y_1(k) = \sum_n x(n)h_1(2k-n)\,. \tag{2.20}$$

The decomposition for one level is illustrated in Figure 2.7b.



(a) (b)

Figure 2.7: (a) Spectral partitioning for QMF filters (b) One-level DWT decomposition.

As can be seen from Figure 2.8, the subband coding process is repeated on the LPF output using the same types of HPF and LPF for further decomposition of the signal until two samples are left. Thus, a data of length 2^J is decomposed into J scales (Mallat algorithm).

At every level (scale), the width of the lower band is halved by the LPF (i.e., its frequency resolution is improved by a factor of two) and its time resolution degrades by two, due to decimation by two [6]. Figure 2.9 shows the resulting spectral partitioning. The frequency components that are most dominant in the signal will have large DWT coefficients, which are located at the scale which includes the particular frequencies. The scales with small DWT coefficients can be discarded without major loss of information.

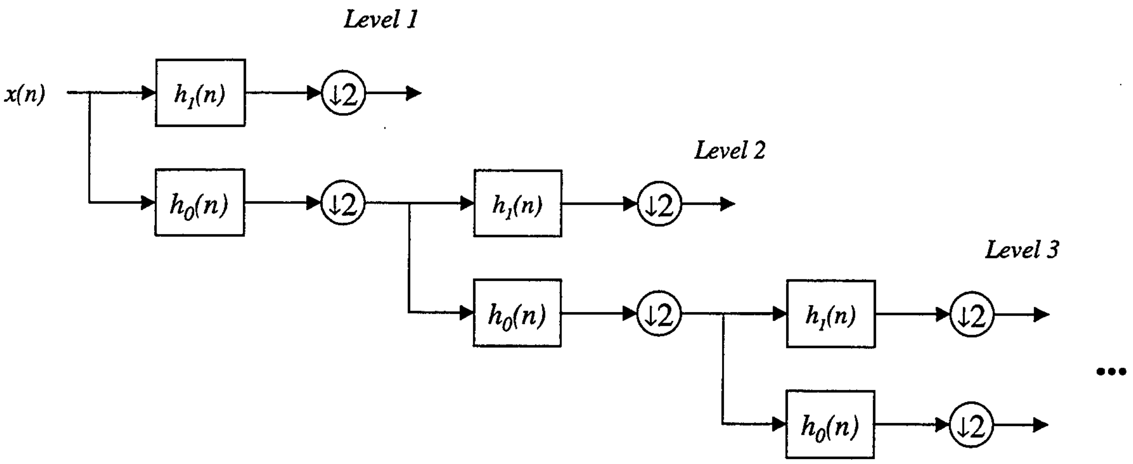


Figure 2.8: Multiple level decomposition.

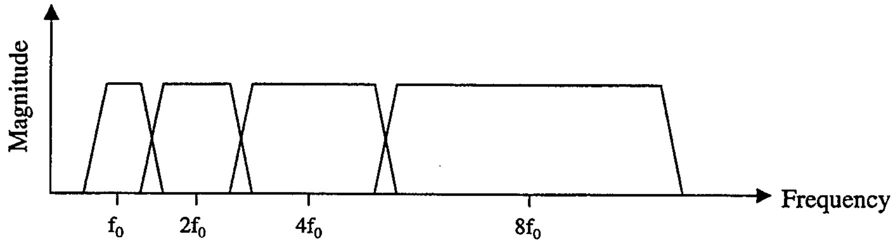


Figure 2.9: Spectral partitioning for the DWT. From Ref. [6].

High frequencies will need more samples while fewer samples are required for low frequencies. The time resolution decreases as the frequency decreases but the frequency resolution increases. In other words, we have good time resolution at high frequencies and good frequency resolution at low frequencies.

The reconstruction of the original signal *x(n)* from its DWT coefficients is called the *synthesis* operation. It is performed by following the above decomposition process in reverse order as illustrated in Figure 2.10. The signals at every level are first interpolated by two (i.e., a zero inserted between each point), passed through the synthesis filters and then added. Thus:

$$\hat{x}(n) = \sum_{k=-\infty}^{\infty} [y_0(k)h_0(2k-n) + y_1(k)h_1(2k-n)]. \tag{2.21}$$

The synthesis filter impulse responses are the time-reversed versions of the analysis filter impulse responses such as $f_0(n)=h_0(P-1-n)$ and $f_1(n)=h_1(P-1-n)$.

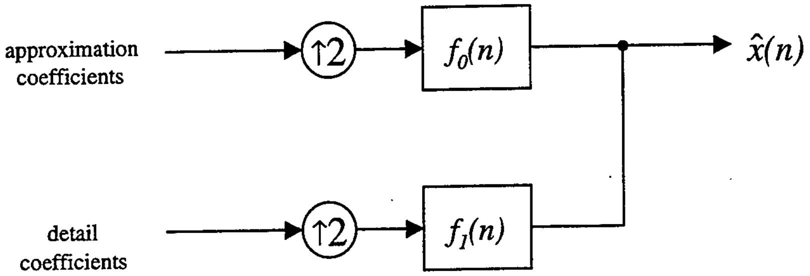


Figure 2.10: Signal reconstruction.

In general, the analysis section of the DWT introduces *aliasing* due to the decimation operation. Some amplitude and phase distortions are also possible during the analysis step. Therefore, for perfect reconstruction and alias cancellation, the synthesis filters must meet certain mathematical conditions and be adapted to the analysis filters [9].

A two channel filter bank yields perfect reconstruction when [9]:

$$F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l}, \tag{2.22}$$

$$F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0, \tag{2.23}$$

where l is the delay in the z-domain. Equation (2.22) is required for no amplitude and phase distortion and Equation (2.23) is required for alias cancellation. Thus, choosing $F_0(z)=H_1(-z)$ and $F_1(z)=-H_0(-z)$ satisfies Equation (2.23), and removes aliasing [9]. Equation (2.23) and a case study related to alias cancellation will be discussed further in Chapter IV.

III. SIGNAL DENOISING

The denoising schemes investigated in this work involve the Wiener filtering of the DWT coefficients. The noise removal performance of each scheme is compared to that of the traditional Wiener filter and to wavelet thresholding. This chapter briefly discusses the Wiener filtering and reviews wavelet-based denoising. It also defines the characteristics of the noise model used in this study.

A. WIENER FILTERING

In a general linear signal estimation problem, which is illustrated in Figure 3.1, the Wiener filter is the optimum linear filter for estimating a signal in additive noise [10]. We consider here the finite impulse response (FIR) form of the Wiener filter of length P.

The noisy observation sequence is represented by:

$$x(n) = s(n) + \eta(n)\,, \tag{3.1}$$

where $s(n)$ is the deterministic signal and $\eta(n)$ is the noise.

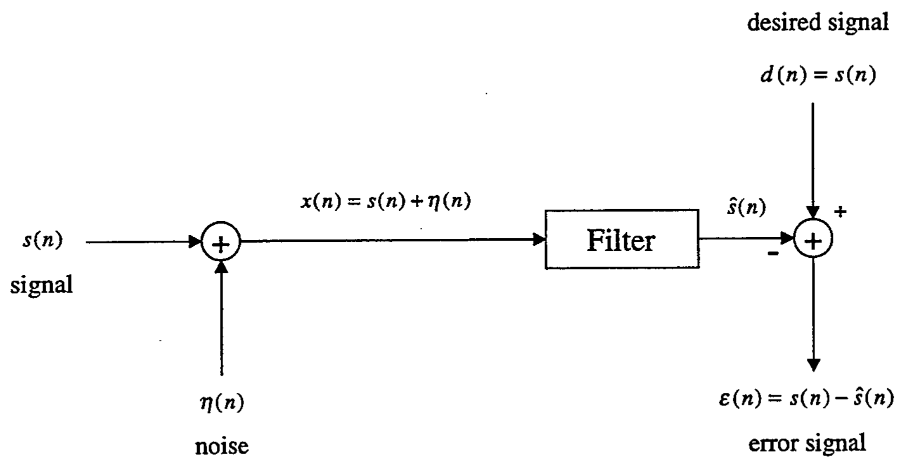


Figure 3.1: General linear signal estimation problem.

The mean-square error (MSE) between the signal and the estimate for the signal is given by:

$$\sigma_\varepsilon^2 = E\left\{\left|\varepsilon(n)\right|^2\right\} = E\left\{\left|s(n) - \hat{s}(n)\right|^2\right\}, \tag{3.2}$$

where $\hat{s}(n)$ is the output of the filter. The filter criterion selected is the minimization of the MSE using the statistical characteristics for the signal and noise [10]. Usually, these statistics are not known a priori and must be estimated from the observed signal *x(n)*.

For a stationary process the output of the filter can be written as:

$$\hat{s}(n) = \sum_{l=0}^{P-1} h(l)x(n-l), \tag{3.3}$$

where *h(l)* is the impulse response of the filter. Consequently the solution is obtained by solving the *Wiener-Hopf* equation, which has the form:

$$\sum_{l=0}^{P-1} R_x(l-i)h^*(l) = R_{sx}^*(i) \qquad ; \qquad i = 0,1,\ldots,P-1 \quad , \tag{3.4}$$

where $R_x(l)$ is the auto-correlation function of the observations, "*" denotes conjugation, and $R_{sx}(l)$ is the cross-correlation function between the desired filter output *s(n)* and the filter input *x(n)* [2]. If the signal and noise are uncorrelated, assuming a zero mean noise, it can be shown that:

$$R_x(l) = R_s(l) + R_\eta(l), \tag{3.5}$$

$$R_{sx}(l) = R_s(l), \tag{3.6}$$

and:

$$R_{sx}(l) = R_x(l) - R_\eta(l), \tag{3.7}$$

where $R_s(l)$ and $R_\eta(l)$ are the auto-correlation functions of the signal and noise, respectively. An estimate for $R_\eta(l)$ can be obtained from the received data prior to the onset of $s(n)$. The MSE is:

$$\sigma_\varepsilon^2 = R_s(0) - \sum_{l=0}^{P-1} h^*(l) R_{sx}(l) . \qquad (3.8)$$

It is easier to derive the *Wiener-Hopf* equation using matrix notations. In this case, the filter output is expressed as:

$$\hat{s}(n) = \sum_{l=0}^{P-1} h(l) x(n-l) = [\tilde{\mathbf{x}}(n)]^T \mathbf{h} , \qquad (3.9)$$

where **h** is defined as:

$$\mathbf{h} = \begin{bmatrix} h(0) \\ h(1) \\ \cdot \\ \cdot \\ \cdot \\ h(P-1) \end{bmatrix} , \qquad (3.10)$$

and:

$$\mathbf{x}(n) = \begin{bmatrix} x(n-P+1) \\ x(n-P) \\ \cdot \\ \cdot \\ \cdot \\ x(n) \end{bmatrix} \quad ; \quad \tilde{\mathbf{x}}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \cdot \\ \cdot \\ \cdot \\ x(n-P+1) \end{bmatrix} . \qquad (3.11)$$

The matrix form of the Wiener-Hopf equation takes the form [2]:

$$\tilde{\mathbf{R}}_\mathbf{x} \mathbf{h}^* = \tilde{\mathbf{r}}_{sx}^* . \qquad (3.12)$$

In the stationary case, $\tilde{\mathbf{R}}_\mathbf{x}^* = \mathbf{R}_\mathbf{x}$ and hence Equation (3.12) can be written as:

$$\mathbf{R}_\mathbf{x} \mathbf{h} = \tilde{\mathbf{r}}_{sx} . \qquad (3.13)$$

The solution for the optimum filter coefficients is:

$$\mathbf{h}_{opt} = \mathbf{R}_{\mathbf{x}}^{-1}\tilde{\mathbf{r}}_{sx}^{*}, \tag{3.14}$$

while the MSE is given by:

$$\sigma_{\varepsilon}^{2} = R_{s}(0) - \tilde{\mathbf{r}}_{sx}^{T}\mathbf{h}^{*} = R_{s}(0) - \mathbf{h}^{*T}\tilde{\mathbf{r}}_{sx}, \tag{3.15}$$

where $\mathbf{R}_{\mathbf{x}} = E\{\mathbf{x}(n)\mathbf{x}^{*T}(n)\}$ and $\mathbf{r}_{sx} = E\{s(n)[\mathbf{x}(n)]^{*}\}$. When the signal and noise are uncorrelated and have zero mean then $\mathbf{R_x}=\mathbf{R_s}+\mathbf{R_\eta}$. Thus, the signal correlation matrix $\mathbf{R_s}$ can be determined from $\mathbf{R_s}=\mathbf{R_x}-\mathbf{R_\eta}$ and $\tilde{\mathbf{r}}_{sx}$ can be extracted from the first column of $\mathbf{R_s}$.

When the signal *s(n)* is non-stationary, the Wiener filter becomes time-varying and take on the form *h(n,l)* [2]. Although the same equations can be used to determine the optimum filter weights, a short-time Wiener filtering algorithm is a more efficient approach to this problem [11].

The Wiener filter procedure is illustrated in the following graphs. Figure 3.2 plots the application of the Wiener filter to a noisy sinusoid of length N=1024 with a constant frequency of f=0.05 at an SNR level of -3 dB. A Wiener filter length of P=30 was chosen. The top three plots display the original, noisy and denoised signals, respectively. The Wiener filter, as shown in the bottom plot of the figure, enhances the frequency bands where the signal level is strong and attenuates the frequency bands where noise is present. Note that the frequency domain plots have a normalized frequency axis.

The performance of the Wiener filter in terms of the normalized MSE is displayed in Figure 3.3 for various combinations of filter and data lengths, normalized signal frequency, and SNR levels. The normalized MSE that will be used for all comparisons is defined as:

$$MSE = \sum_{n=1}^{N}\left(\frac{s(n)}{\|s(n)\|} - \frac{\hat{s}(n)}{\|\hat{s}(n)\|}\right)^{2}, \tag{3.16}$$

where N, $s(n)$ and $\hat{s}(n)$ respectively represent the data length, the noise free and denoised signal. Thus, the MSE chosen compares the signals which are energy normalized.

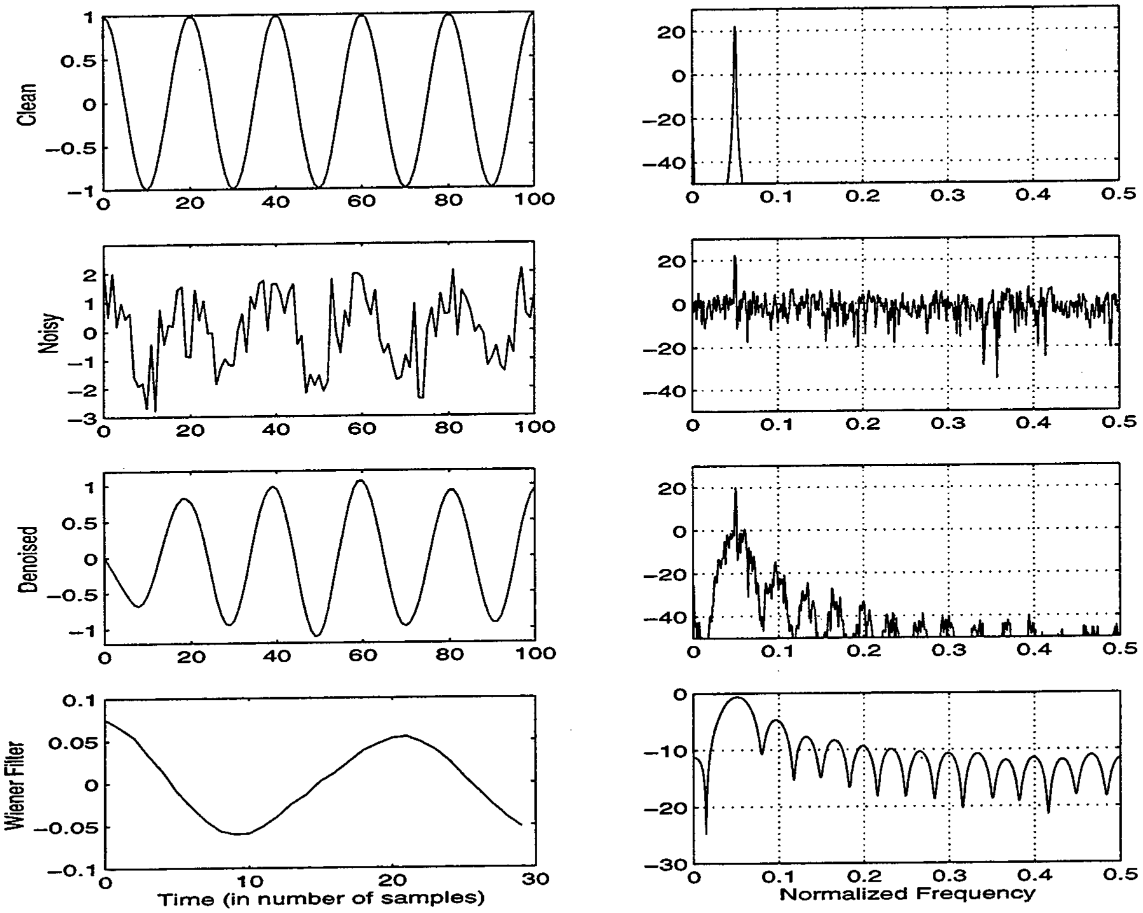


Figure 3.2: Denoising with the Wiener filtering. Sampling frequency $f_s=1$. Left column: time domain signal. Right column: spectrum in dB.

The top plots display MSE versus SNR levels between −15 to 15 dB. The bottom plots display MSE versus the normalized frequency spanning the range from 0 to 0.5. The graphs represent the average MSE values obtained from ten independent trials. From the left-hand side plots of this figure, it is clear that the MSE decreases as the filter length increases. In addition, the performances of the filter orders 12, 20 and 30 are close to each other.

A data size of 1000 or more points was used to obtain reliable estimates of the correlation matrices [1]. However, the right-hand side plots of Figure 3.3 indicate that increasing the length of the observed sequence does not significantly improve the MSE. As a result, it seems reasonable to choose *P=12* and *N=1024* from the viewpoint of the computational cost factor. Thus, these values will be used later in the performance comparisons of the denoising algorithms investigated in this study.

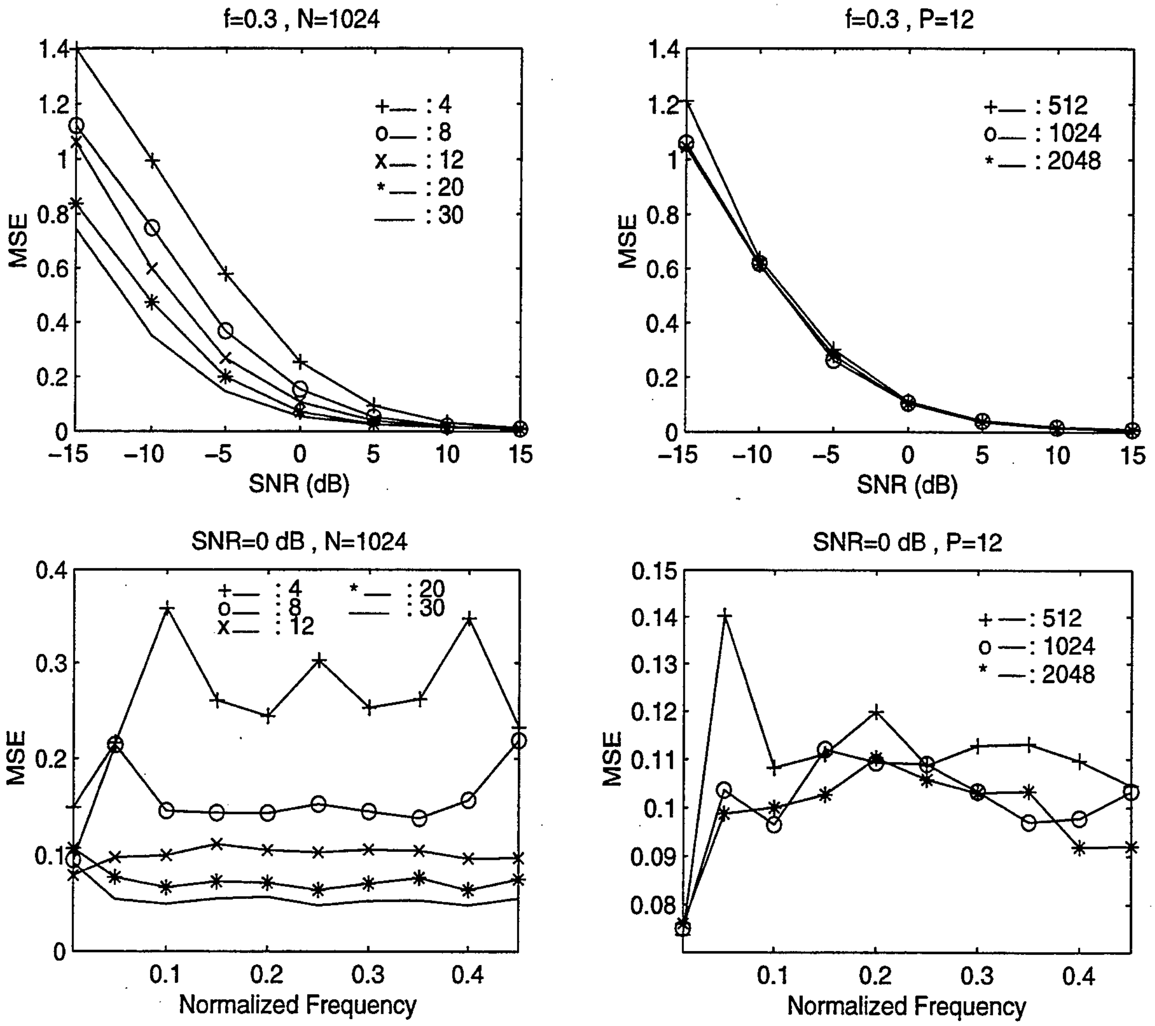


Figure 3.3: The performance of the Wiener filter. The left two plots are for various filter lengths. The right two plots are for various data lengths.

B. WAVELET DENOISING

As mentioned earlier, wavelet signal decompositions have numerous applications in signal processing including noise suppression. It has been proven that denoising in the wavelet domain is often much easier than in the other conventional domains. Moreover, it preserves the sharp features of the signal to be recovered, unlike the traditional techniques which use convolutional filtering [3].

Wavelet shrinkage, introduced by Donoho and Johnstone [12,13,14], is a nonlinear processing technique that can be employed for data compression or denoising according to the *thresholding* method used. Recall that in the wavelet decomposition, each transform coefficient can be viewed as a measure of the correlation between the signal and decomposing basis. Thus, the transform will contain a few large coefficients for good correlation and many small coefficients for poor correlation. Although all the coefficients are required for perfect reconstruction, the general shape of the original signal can be represented by a small number of coefficients carrying significant signal energy (i.e., coefficients with large magnitudes).

Wavelet shrinkage algorithm keeps only those important coefficients based on a nonlinear thresholding. It is a discarding operation of the coefficients that fall below a given magnitude. Wavelet denoising consists of three steps [7], namely:

1- Taking the wavelet transform of the data by decomposing it into J levels with a suitable wavelet basis function,

2- Selecting a threshold T for each level from 1 to J and *soft* or *hard* thresholding the wavelet (detail) coefficients, 3- Performing inverse wavelet transform to reconstruct an estimate of the underlying signal using the original approximation coefficients and the modified detail coefficients.

Hard thresholding zeroes out the coefficients $W_{J,k}$ of magnitudes lower than the threshold value and is given by:

$$W_{J,k}^{H} = \begin{cases} W_{J,k}, & |W_{J,k}| > T \\ 0, & |W_{J,k}| \leq T \end{cases} . \qquad (3.17)$$

Soft thresholding is an extension of the hard thresholding is given by:

$$W_{J,k}^{S} = \begin{cases} sign(W_{J,k})(|W_{J,k}| - T), & |W_{J,k}| > T \\ 0, & |W_{J,k}| \leq T \end{cases} . \qquad (3.18)$$

It is clear from Equation (3.18) that soft thresholding first zeroes out the coefficients of magnitudes smaller than the threshold value, and next shrink the magnitude of the remaining coefficients towards zero. Note that $W_{J,k}^{H}$ has discontinuities while $W_{J,k}^{S}$ does not.

Both thresholding methods mentioned above may be used in denoising and data compression. In general, hard thresholding is used in data compression while soft thresholding is usually employed for denoising.

The noisy data can be expressed as:

$$x(n) = s(n) + \sigma w(n), \qquad (3.19)$$

where $s(n)$ is the signal to be recovered from the noisy data, $w(n)$ is zero mean additive white Gaussian noise (AWGN) and σ is the standard deviation of the AWGN. Since the orthogonal wavelet transforms are linear operations, the wavelet coefficients will be in the form of Equation (3.19).

In the *J*-level wavelet decomposition of *x(n)*, the approximation coefficients at every successive level become less and less noisy. The high frequency components in the signal due to the noise show up in the detail coefficients and mostly at the finest scale [7]. The finest scale is the frequency band between $\pi/2$ and π, where π corresponds to the half sampling frequency. The goal of the wavelet-based denoising is to retain the coefficients, which contain signal information while removing those due to the noise contribution. Thresholding is a lossy operation and usually provides a lowpass version of the original signal. Thus, it cannot provide perfect reconstruction of the original signal since thresholding the detail coefficients will also remove some of the signal components [9].

Algorithms computing the threshold value T require an estimation of σ. Several variants include Stein's Unbiased Risk Estimate (SURE), fixed threshold, Minimax criteria or a combination of those. The Matlab Wavelet Toolbox from MathWorks used in this study employs four thresholding methods, in its routine function *thselect*, as follows [7]:

1- *Rigrsure*: Adaptive threshold selection using SURE,

2- *Sqtwolog*: Fixed form threshold equal to $\sqrt{2\log(N)}$,

3- *Heursure*: A mixture of the previous options,

4- *Minimaxi*: Fixed threshold for achieving minimax performance in terms of the MSE,

where N is the number of wavelet coefficients.

Results show that wavelet denoising performances improve as the SNR level increases and the frequency decreases. We also noted that thresholding schemes fail at SNR levels 0 dB and below.

The MSE performance comparisons of the threshold selection methods of the toolbox are displayed in Figure 3.4. The test signal is a cosine wave of 1024 points duration with a constant frequency of 0.3 at a sampling frequency of 1. As mentioned in the DWT section of Chapter II, this signal can be decomposed into 10 levels. However, a 5-level wavelet decomposition covers almost the complete frequency spectrum except very low frequencies. The *sym8* (i.e., *Symlet8*) wavelet was empirically chosen and the signal was decomposed into 5-level wavelet coefficients. For every level, a threshold value calculated according to the given threshold selection method and soft thresholding was applied to the coefficients. The resulting MSE indexed by SNR is shown in the top part of Figure 3.4. The bottom part of the figure plots MSE versus the normalized frequency at a fixed SNR level of 0 dB. All points represent averages of 10 independent runs. Without loss of generality, it can be seen that the accuracy of all four methods decreased as the frequency increased. As expected, the error of all techniques decreased as the SNR increased. From both plots, it is clear that the fixed form threshold has the worst performance. The reason for this might be due to the fact that the threshold obtained by *sqtwolog* was set too large to keep all the signal information coefficients. On the other hand, the *rigrsure* and *heursure* methods attained the best performance for all conditions.

The soft and hard thresholding methods are compared in Figure 3.5 where the *rigrsure* wavelet thresholding method is applied to the same noisy tonal used in the previous section. The signal was decomposed into 5 levels using the *sym8* wavelet. We note that soft thresholding leads to a better performance in the MSE sense.

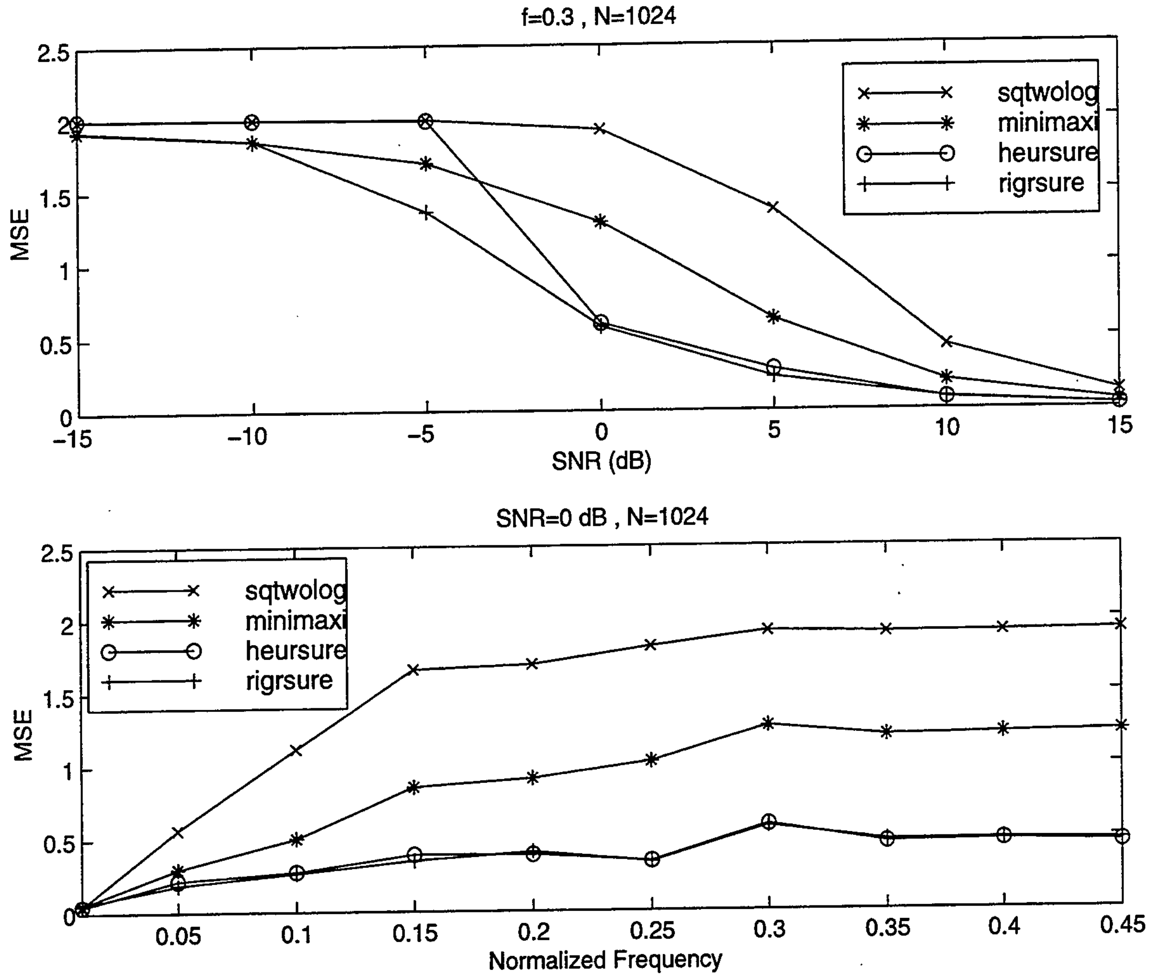


Figure 3.4: Comparison of different threshold selection methods.

We noted that wavelet denoising schemes usually threshold the detail coefficients only, as done for example in the Wavelet toolbox. However, we also noted that, in cases where the signal information is not contained in the approximations, denoising performances were improved by thresholding the approximation coefficients, too. This is to be expected as in such cases the approximation coefficients contain noise only, and as a result should be disregarded.

Finally, some knowledge about the signal in advance is useful in noise reduction using wavelet thresholding, since choosing a suitable wavelet basis function that matches the signal character will provide a few but large magnitude wavelet coefficients that represent good correlation.

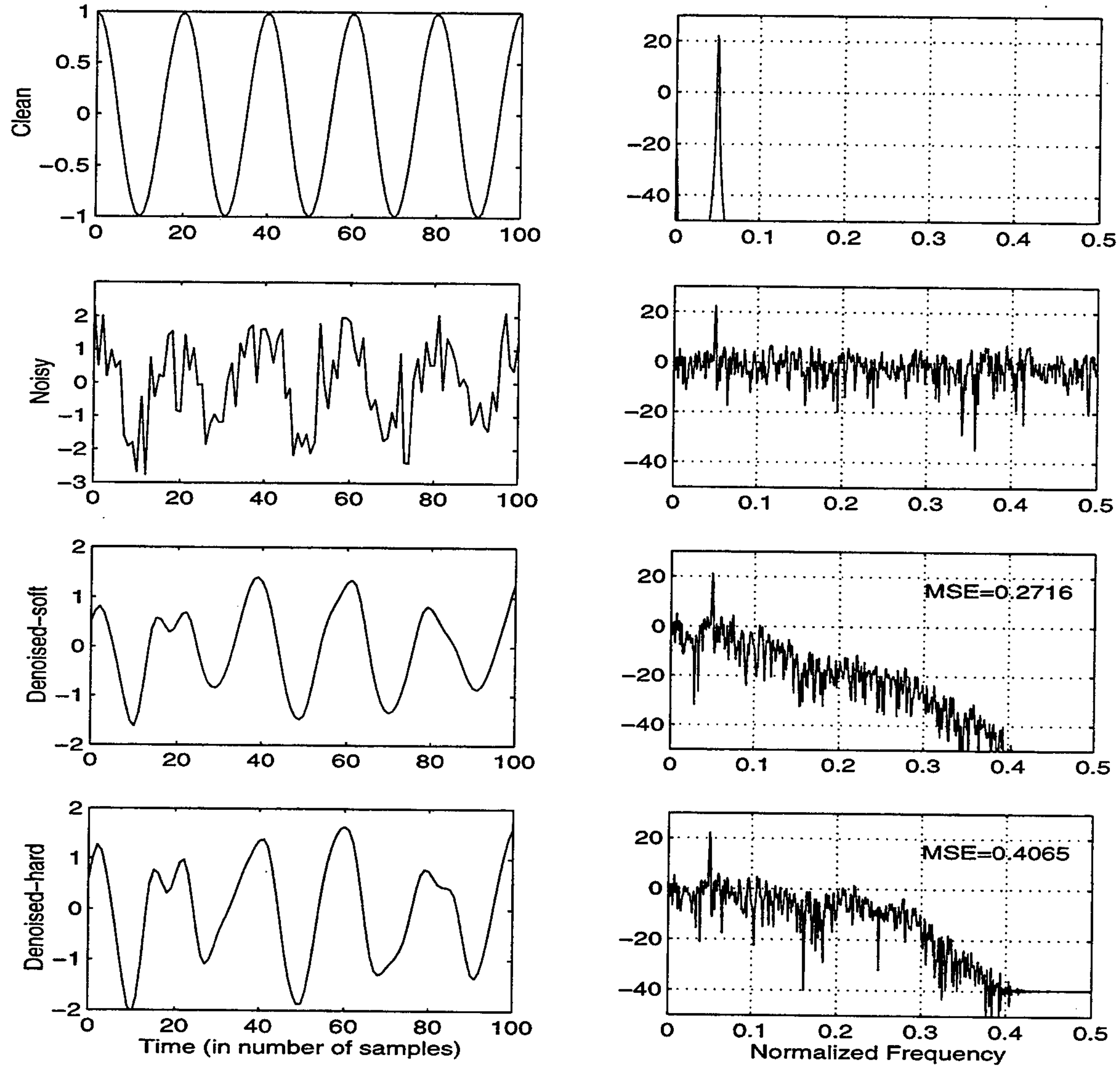


Figure 3.5: Denoising with the wavelet thresholding. Sampling frequency $f_s=1$. Left column: time domain signal. Right column: spectrum in dB.

C. NOISE DESCRIPTION

It was desirable to make a few assumptions of the additive noise model used to provide simplicity in the study and analysis of the denoising algorithms that we investigated in this thesis.

Results have shown that the ocean acoustic noise is usually a colored Gaussian random process [1]. However, in this study it was assumed to be zero mean additive white Gaussian noise (AWGN) based on the fact that some pre-whitening techniques can be employed if necessary. This assumption was very helpful to allow a wide variety of simulations for different underwater signals and comparisons using different filtering techniques.

The noise was assumed to be a stationary process. This assumption holds as generally denoising algorithms are concerned with short durations of data of a few seconds. In such short periods of time, the properties of the ocean itself and the ambient noise change very little [1].

Next, we assumed that the variance of the AWGN was unity and varied the signal amplitude to obtain various SNR levels. This also enabled us to make comparisons of our algorithms to those of Donoho's study.

Finally, we note that our results are based on the knowledge of the exact noise-only sequence. In practical applications, one does not have access to the exact noise information which then must be estimated. These estimates are expected to lead to performance degradations in the various schemes studied.

IV. WAVELET/WIENER-BASED DENOISING

The previous chapter described how Wiener filtering and wavelet thresholding can be applied to remove noise from acoustic signals. In this chapter, we will present four denoising algorithms which combine the DWT and Wiener filter. Our goal is to investigate whether these combinations can achieve a better performance than those obtained with either the classic Wiener filter or wavelet thresholding in a MSE sense. The main idea behind the schemes investigated is to apply FIR Wiener filtering to the data transformed to the wavelet domain.

A. DENOISING USING ORTHOGONAL DECIMATED WAVELET TRANSFORMS

Consider the following 2-channel multirate system, which is essentially a QMF analysis/synthesis bank with additional Wiener filtering operations, represented by W_0 and W_1, inserted in each bank. Recall that the original QMF setup is alias-free.

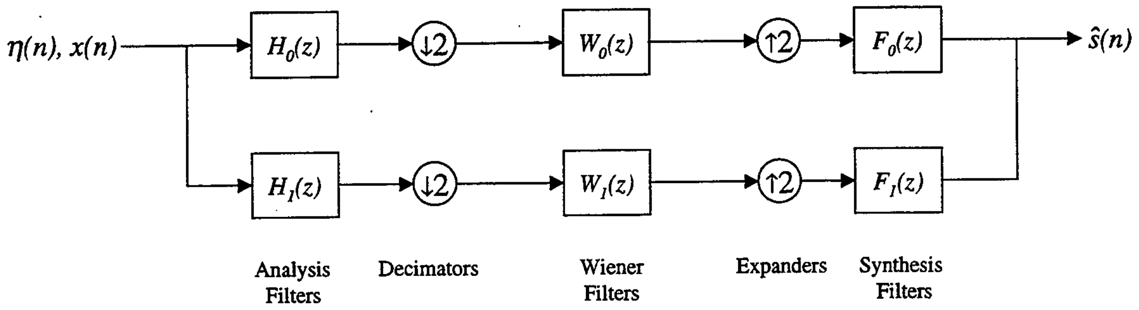


Figure 4.1: Decimated Wavelet/Wiener-based denoising, *1-level* decomposition.

The system input is the noisy signal $x(n)=s(n)+\eta(n)$, where $s(n)$ and $\eta(n)$ are the deterministic signal and the noise, respectively. Given that we have access to a noise only segment of noisy data, the one-level DWT is first applied to this noise-only segment. Next, the DWT of the noisy data is computed. The correlation function of the wavelet filtered noise and noisy signal determine the characteristics of the Wiener filter. The outputs of the Wiener filters represent the modified approximation and detail coefficients, which will be used for the reconstruction to get an estimation of the clean signal $s(n)$, as shown in Figure 4.2.

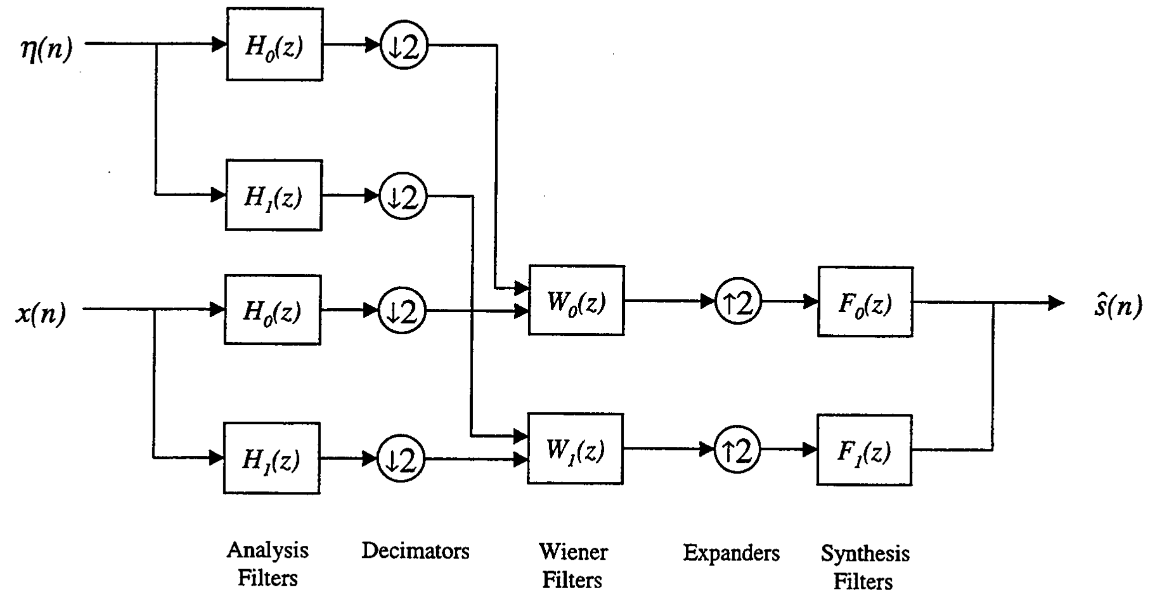


Figure 4.2: Wavelet/Wiener-based denoising scheme.

The denoising operation can be easily expanded to multiple levels using the orthogonal DWT decomposition. Figure 4.3 illustrates the J-level Wavelet/Wiener filtering procedure.

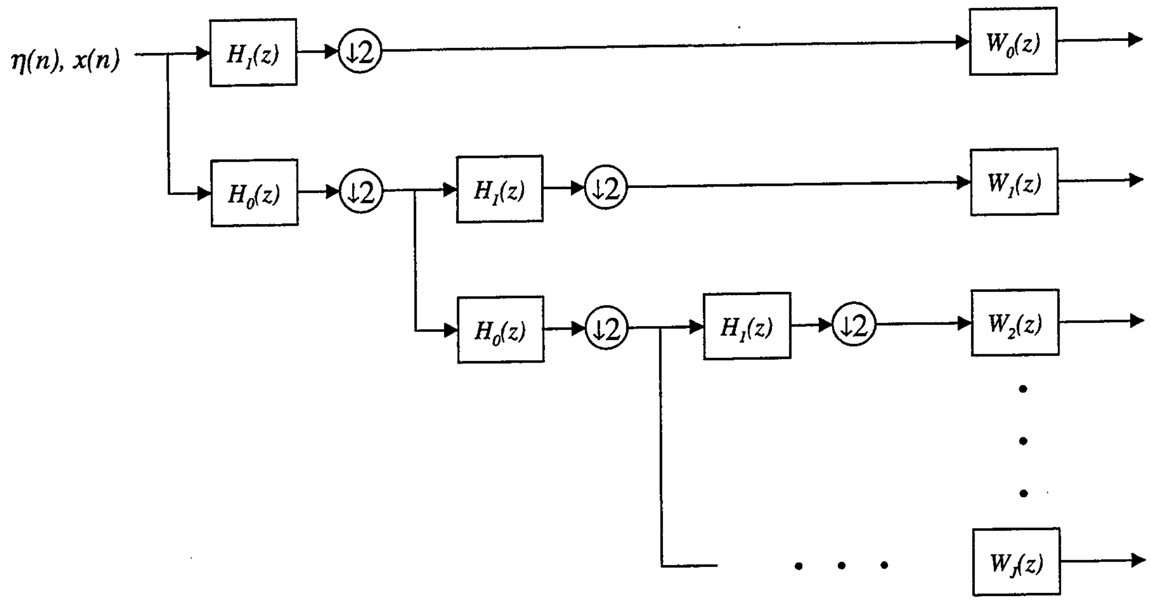


Figure 4.3: The analysis step of the *J-level* Wavelet/Wiener-based denoising.

Note that in a perfect reconstruction QMF bank, the reconstructed signal is merely a scaled and delayed version of the input. However, introducing Wiener filters in the above system produces aliasing and distortion, as the analysis/synthesis relationship is modified. Note that aliasing is undesirable and must be canceled to achieve a good estimation of the signal *s(n)*. The next sections will present different approaches in order to overcome this problem.

Figure 4.4 depicts the application of the Wavelet/Wiener scheme to the noisy sinusoid already considered in the previous chapter. A Wiener filter length of 12 and a 5-level decomposition is used. The first two top plots show the noisy signal and the resulting Wiener filtered signal. Denoised signals obtained with the combined method are shown in the third and fourth plots, where *db10* (i.e., *Daubechies10*) and *db40* wavelets are used, respectively.

Note that choosing a higher order wavelet reduces aliasing effects and the MSE, as well. Moreover, the Wavelet/Wiener denoising gives a better performance than the usual Wiener filtering in a MSE sense. However, they also both show the presence of aliasing in the denoised signal.

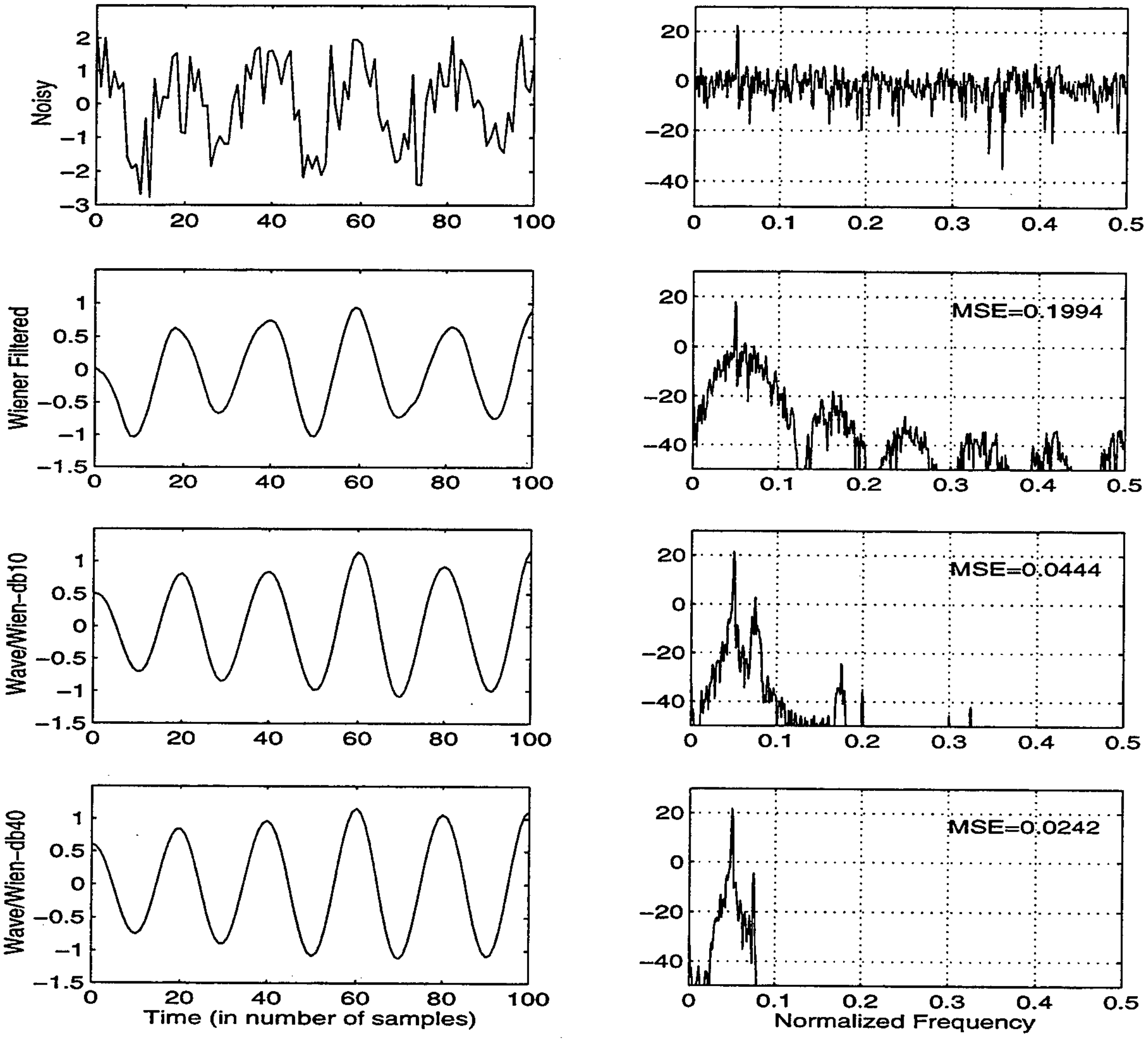


Figure 4.4: Denoising with the Wavelet/Wiener method. Sampling frequency is 1. Left column: time domain signal. Right column: spectrum in dB.

Figure 4.5 displays the spectrum of the tonal at normalized frequency equal to 0.3. For this frequency the Wiener filter gives a MSE of 0.1833. Although both spectra indicate aliasing, they have still better MSEs than those obtained using the Wiener filter.

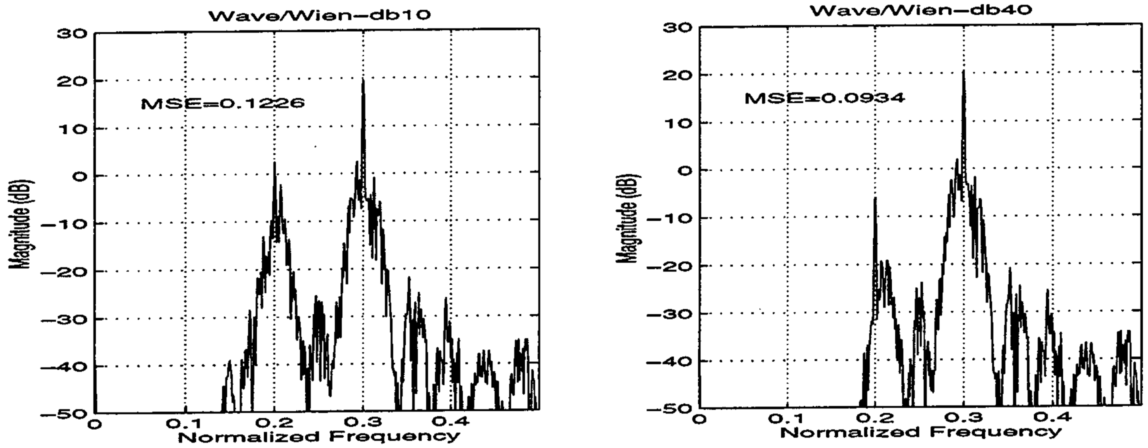


Figure 4.5: Denoising with the Wavelet/Wiener method using *db10* and *db40* wavelets. *5-level* wavelet decomposition.

B. DENOISING USING *1*-LEVEL ORTHOGONAL UNDECIMATED WAVELET TRANSFORMS

We showed in the previous section that introducing the frequency band Wiener filters resulted in creating aliasing in the denoised signal. This section investigates a *1*-level high-pass/low-pass (i.e., band-specific) Wiener filter denoising scheme in the wavelet domain, where the decimation and interpolation operations have been removed, as illustrated in Figure 4.6. The main idea is to compare the band-specific Wiener filter to the classic Wiener filter in the absence of further distortions created by the decimation/interpolation operations present in the decimated wavelet transform.

Simulations showed that this implementation introduces amplitude scaling in the filtered signal due to successive filtering operations. To allow visual comparison we normalized the output signals to have unit energy.

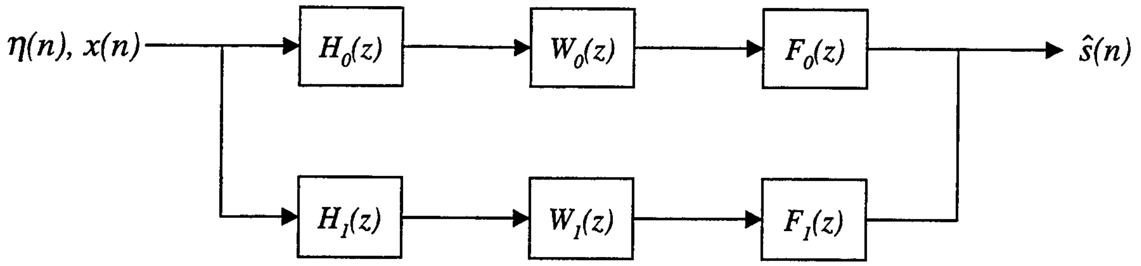


Figure 4.6: Undecimated Wavelet/Wiener-based denoising.

Figure 4.7 illustrates the noisy, band-specific Wiener filtered signals using two different wavelets (i.e., *db10* and *db40*) and the respective spectra obtained for the single tone at normalized frequency 0.05 used previously in Section A. The length of the Wiener filter is 12 and SNR=-3 dB. The plots show that no aliasing or distortion is present. As can be seen the MSE values are higher than those of the scheme with decimators and expanders, however, they are still lower than that of the classic Wiener filter.

Figure 4.8 compares the spectra of the denoised signals obtained with the Wiener filter, the decimated Wavelet/Wiener, and the undecimated Wavelet/Wiener denoising schemes using a single tone at normalized frequency equal to 0.3 and an SNR level of –3 dB. The Wiener filter length is 12 and the signal is decomposed into 5 levels in the decimated Wavelet/Wiener filter scheme.

Results show that increasing the wavelet order in the undecimated Wavelet/Wiener scheme does not seem to improve the MSE performance significantly.

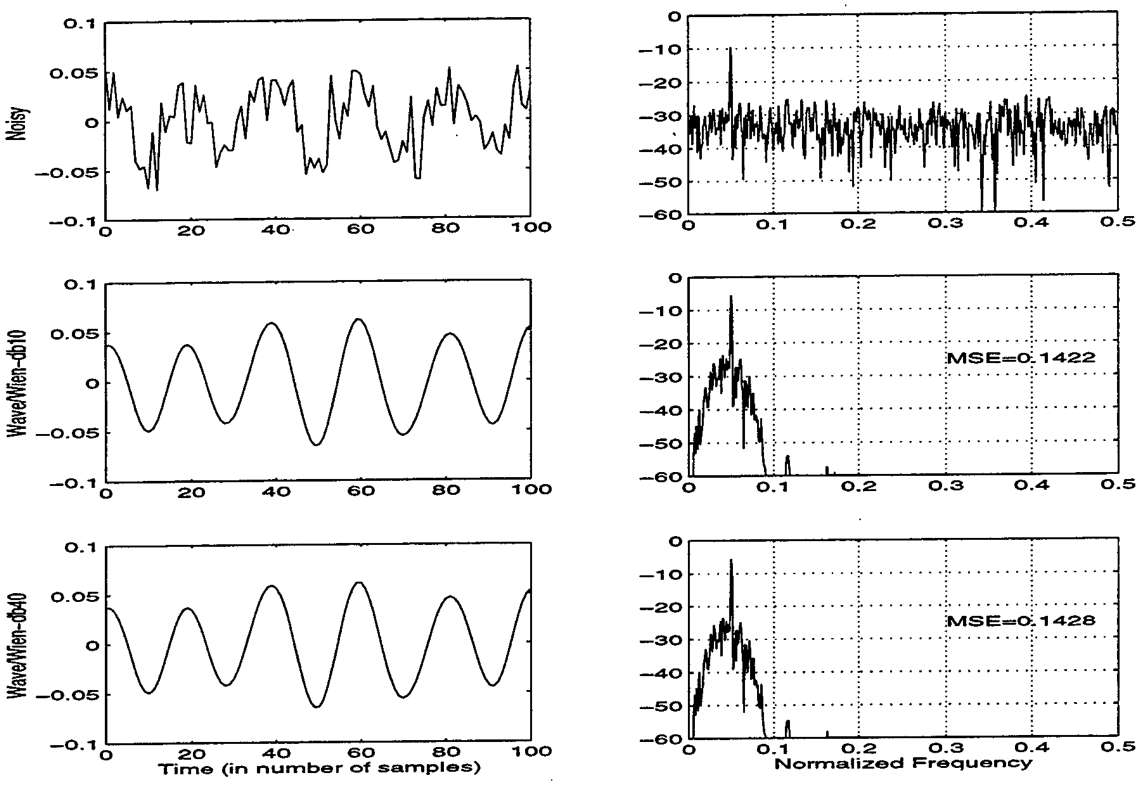


Figure 4.7: Undecimated Wavelet/Wiener Denoising scheme. Left column: time domain signal normalized to unit energy. Right column: spectrum in dB.

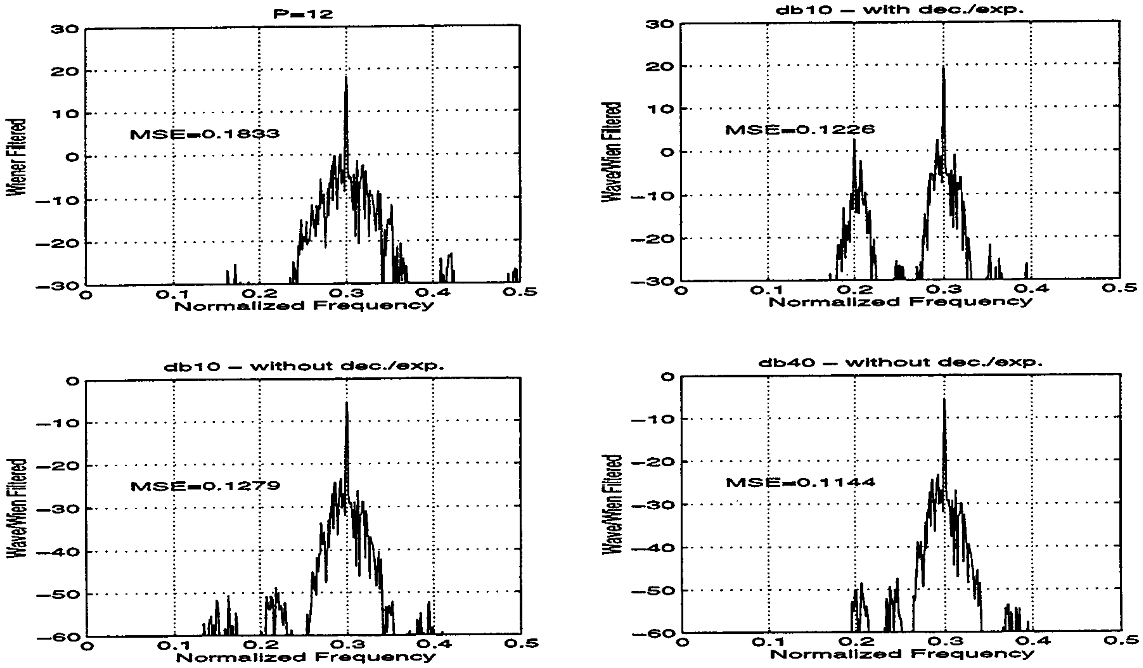


Figure 4.8: Filtering scheme comparisons. One single tone at normalized frequency f=0.3 and SNR=-3 dB.

C. DENOISING USING ORTHOGONAL DECIMATED WAVELET TRANSFORMS WITH ALIAS CANCELLATION

We showed in previous sections that aliasing present in the decimated Wavelet/Wiener filter scheme is a direct result of the decimation/expansion operations present in the decimated wavelet transform. In this section, we consider two schemes designed to cancel the aliasing present in the denoised signal when applying the decimated transform. The first one attempts to modify each wavelet synthesis filter independently of the others, while the second one uses cross-band filters.

1. Alias Cancellation Synthesis Filters

Consider the following *1*-level decimated Wavelet/Wiener denoising scheme where the filters $G_k(z)$ $(k=0,1)$ are introduced for alias cancellation.

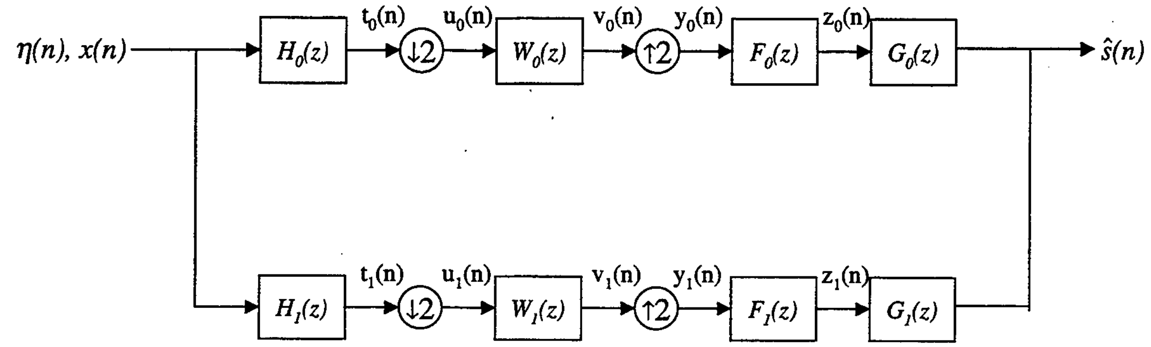


Figure 4.9: *1-level* decimated Wavelet/Wiener filter scheme with alias cancellation filters.

The filters $G_k(z)$ are to be defined so that the denoised signal $\hat{s}(n)$ contains no aliasing term.

The z-transform of the outputs to the analysis filters H_0 and H_1 are given by:

$$T_0(z)=H_0(z)X(z),$$

$$T_1(z)=H_1(z)X(z). \quad (4.1)$$

The z-transforms of the decimated signals $u_k(n)$ are given by:

$$U_0(z)=1/2[H_0(z^{1/2})X(z^{1/2})+H_0(-z^{1/2})X(-z^{1/2})],$$

$$U_1(z)=1/2[H_1(z^{1/2})X(z^{1/2})+H_1(-z^{1/2})X(-z^{1/2})]. \quad (4.2)$$

The z-transforms of the Wiener filters' outputs $v_k(n)$ are:

$$V_0(z)=1/2[W_0(z)H_0(z^{1/2})X(z^{1/2})+W_0(z)H_0(-z^{1/2})X(-z^{1/2})],$$

$$V_1(z)=1/2[W_1(z)H_1(z^{1/2})X(z^{1/2})+W_1(z)H_1(-z^{1/2})X(-z^{1/2})]. \quad (4.3)$$

After the interpolation steps, we have:

$$Y_0(z)=V_0(z^2)=1/2[W_0(z^2)H_0(z)X(z)+W_0(z^2)H_0(-z)X(-z)],$$

$$Y_1(z)=V_1(z^2)=1/2[W_1(z^2)H_1(z)X(z)+W_1(z^2)H_1(-z)X(-z)]. \quad (4.4)$$

Next, the z-transforms of the terms $z_k(n)$ are given by:

$$Z_0(z)=1/2[F_0(z)W_0(z^2)H_0(z)X(z)+F_0(z)W_0(z^2)H_0(-z)X(-z)],$$

$$Z_1(z)=1/2[F_1(z)W_1(z^2)H_1(z)X(z)+F_1(z)W_1(z^2)H_1(-z)X(-z)]. \quad (4.5)$$

Finally, the z-transform of the reconstructed signal $\hat{s}(n)$ is:

$$\hat{S}(z)=G_0(z)Z_0(z)+G_1(z)Z_1(z). \quad (4.6)$$

Substituting Equations (4.1) to (4.5) into Equation (4.6) leads to:

$$\hat{S}(z)=1/2X(z)[G_0(z)F_0(z)W_0(z^2)H_0(z)+G_1(z)F_1(z)W_1(z^2)H_1(z)]$$

$$+1/2X(-z)[G_0(z)F_0(z)W_0(z^2)H_0(-z)+G_1(z)F_1(z)W_1(z^2)H_1(-z)]. \quad (4.7)$$

Recall that canceling the alias term $X(-z)$ is required to get perfect reconstruction. It can be seen from Equation (4.7) that the aliasing term can be canceled by choosing the filters such that the term $G_0(z)F_0(z)W_0(z^2)H_0(-z)+G_1(z)F_1(z)W_1(z^2)H_1(-z)$ is equal to zero. Therefore, aliasing is canceled by selecting the filters $F_k(z)$ and $G_k(z)$ as:

$$F_0(z)=H_1(-z) \quad , \quad F_1(z)=-H_0(-z), \tag{4.8}$$

$$G_0(z)=W_1(z^2) \quad , \quad G_1(z)=W_0(z^2). \tag{4.9}$$

The QMF bank perfect reconstruction property already implements Equation (4.8). Equation (4.9) can be implemented after designing the Wiener filters.

Replacing $F_0(z)$, $F_1(z)$, $G_0(z)$ and $G_1(z)$ by their values in Equation (4.7) leads to $\hat{S}(z)=1/2X(z)[W_1(z^2)H_1(-z)W_0(z^2)H_0(z)-W_0(z^2)H_0(-z)W_1(z^2)H_1(z)]$. Using the fact that $H_1(z)=-z^{-N}H_0(-z^{-1})$ leads to $\hat{S}(z)=W_0(z^2)W_1(z^2)z^{-N}X(z)$.

Simulations results showed that this scheme did not give satisfactory results as it introduced further distortions in the denoised signal, even though the aliasing terms tend to be canceled [15]. Thus, the second alternative considered is presented next.

2. Alias Cancellation Analysis Filters

In this section, we consider a decimated Wavelet/Wiener denoising scheme, shown in Figure 4.10, where $K_k(z)$ $(k=0,1)$ are alias cancellation filters whose lengths are the same as those of the Wiener filters $W_k(z)$. As done earlier, the alias cancellation filters will be derived by using the z-transform.

The z-transform of the outputs to the analysis filters H_0 and H_1 are given by:

$$T_0(z)=H_0(z)X(z),$$

$$T_1(z)=H_1(z)X(z). \tag{4.10}$$

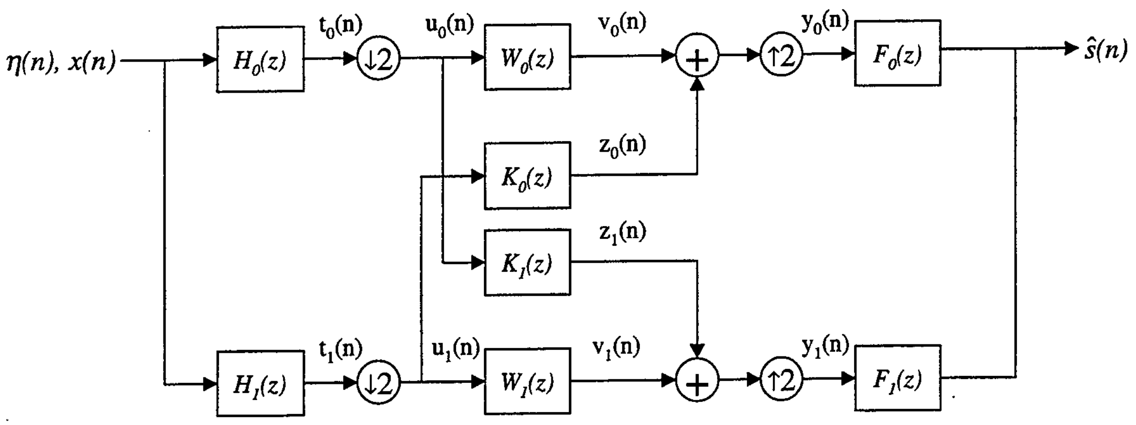


Figure 4.10: *1-level* decimated Wavelet/Wiener filter scheme with alias cancellation analysis filters.

The z-transforms of the decimated signals $u_k(n)$ are given by:

$$U_0(z)=1/2[H_0(z^{1/2})X(z^{1/2})+H_0(-z^{1/2})X(-z^{1/2})],$$

$$U_1(z)=1/2[H_1(z^{1/2})X(z^{1/2})+H_1(-z^{1/2})X(-z^{1/2})]. \qquad (4.11)$$

The z-transforms of the Wiener filters outputs $v_k(n)$ are:

$$V_0(z)=1/2[W_0(z)H_0(z^{1/2})X(z^{1/2})+W_0(z)H_0(-z^{1/2})X(-z^{1/2})],$$

$$V_1(z)=1/2[W_1(z)H_1(z^{1/2})X(z^{1/2})+W_1(z)H_1(-z^{1/2})X(-z^{1/2})]. \qquad (4.12)$$

The z-transforms of the alias cancellation filters outputs $z_k(n)$ are:

$$Z_0(z)=1/2[K_0(z)H_1(z^{1/2})X(z^{1/2})+K_0(z)H_1(-z^{1/2})X(-z^{1/2})],$$

$$Z_1(z)=1/2[K_1(z)H_0(z^{1/2})X(z^{1/2})+K_1(z)H_0(-z^{1/2})X(-z^{1/2})]. \qquad (4.13)$$

After summation and interpolation steps, we have:

$$Y_0(z)=V_0(z^2)+Z_0(z^2)=1/2[W_0(z^2)H_0(z)X(z)+W_0(z^2)H_0(-z)X(-z)]$$
$$+1/2[K_0(z^2)H_1(z)X(z)+K_0(z^2)H_1(-z)X(-z)],$$

$$Y_1(z)=V_1(z^2)+Z_1(z^2)=1/2[W_1(z^2)H_1(z)X(z)+W_1(z^2)H_1(-z)X(-z)]$$
$$+1/2[K_1(z^2)H_0(z)X(z)+K_1(z^2)H_0(-z)X(-z)]. \qquad (4.14)$$

Finally, the z-transform of the reconstructed signal $\hat{s}(n)$ is:

$$\hat{S}(z) = F_0(z)Y_0(z)+F_1(z)Y_1(z). \qquad (4.15)$$

Substituting Equations (4.10) to (4.14) into Equation (4.15) leads to:

$$\begin{aligned}\hat{S}(z) = 1/2X(z)[&F_0(z)W_0(z^2)H_0(z)+F_1(z)W_1(z^2)H_1(z)\\ &+F_0(z)K_0(z^2)H_1(z)+F_1(z)K_1(z^2)H_0(z)],\\ +1/2X(-z)[&F_0(z)W_0(z^2)H_0(-z)+F_1(z)W_1(z^2)H_1(-z)\\ &+F_0(z)K_0(z^2)H_1(-z)+F_1(z)K_1(z^2)H_0(-z)]. \qquad (4.16)\end{aligned}$$

It can be seen from Equation (4.16) that the aliasing term $X(-z)$ can be canceled when:

$$F_0(z)W_0(z^2)H_0(-z)+F_1(z)W_1(z^2)H_1(-z)+F_0(z)K_0(z^2)H_1(-z)+F_1(z)K_1(z^2)H_0(-z)=0. \qquad (4.17)$$

Recall that $F_0(z)=H_1(-z)$ and $F_1(z)=-H_0(-z)$ can be selected to insure the analysis and synthesis filters properties lead to perfect QMF reconstruction. Substituting these properties into Equation (4.17) and rearranging leads to:

$$F_0(z)F_1(z)[W_1(z^2)-W_0(z^2)]+F_0(z)^2K_0(z^2)-F_1(z)^2K_1(z^2)=0, \qquad (4.18)$$

and,

$$F_0(z)F_1(z)[W_1(z^2)-W_0(z^2)]=F_1(z)^2K_1(z^2)-F_0(z)^2K_0(z^2). \qquad (4.19)$$

For a wavelet filter of order of L (i.e., filter length is L+1), Wiener filters and alias cancellation filters of order of P (i.e., filter length is P+1), the filters $F_0(z)$,$F_1(z)$,$W_0(z^2)$,$W_1(z^2)$,$K_0(z^2)$ and $K_1(z^2)$ are given by:

$$F_0(z)=a_0+a_1z^{-1}+\ldots+a_Lz^{-L}. \qquad (4.20)$$

$$F_1(z)=b_0+b_1z^{-1}+\ldots+b_Lz^{-L}. \qquad (4.21)$$

$$W_0(z^2)=m_0+m_1z^{-2}+\ldots+m_Pz^{-2P}. \qquad (4.22)$$

$$W_1(z^2)=n_0+n_1z^{-2}+\ldots+n_Pz^{-2P}. \qquad (4.23)$$

$$K_0(z^2)=y_0+y_1z^{-2}+\ldots+y_Pz^{-2P}. \tag{4.24}$$

$$K_1(z^2)=z_0+z_1z^{-2}+\ldots+z_Pz^{-2P}. \tag{4.25}$$

Thus,

$$\begin{aligned} F_0(z)F_1(z)&=a_0b_0+\ (a_0b_1+a_1b_0)z^{-1}+\ldots+a_Lb_Lz^{-2L} \\ &=h_0+h_1z^{-1}+\ldots+h_{2L}z^{-2L}, \end{aligned} \tag{4.26}$$

where $h_0=a_0b_0,\ h_1=a_0b_1+a_1b_0,\ldots,\ h_{2L}=a_Lb_L$.

$$\begin{aligned} F_0(z)^2&=F_0(z)F_0(z)=(a_0+a_1z^{-1}+\ldots+a_Lz^{-L}).(a_0+a_1z^{-1}+\ldots+a_Lz^{-L}) \\ &=a_0a_0+(a_0a_1+a_1a_0)z^{-1}+\ldots+a_La_Lz^{-2L}=k_0+k_1z^{-1}+\ldots+k_{2L}z^{-2L}, \end{aligned} \tag{4.27}$$

where $k_0=a_0a_0,\ k_1=a_0a_1+a_1a_0,\ldots,\ k_{2L}=a_La_L$.

$$\begin{aligned} F_1(z)^2&=F_1(z)F_1(z)=(b_0+b_1z^{-1}+\ldots+b_Lz^{-L}).(b_0+b_1z^{-1}+\ldots+b_Lz^{-L}) \\ &=b_0b_0+(b_0b_1+b_1b_0)z^{-1}+\ldots+b_Lb_Lz^{-2L}=g_0+g_1z^{-1}+\ldots+g_{2L}z^{-2L}, \end{aligned} \tag{4.28}$$

where $g_0=b_0b_0,\ g_1=b_0b_1+b_1b_0,\ldots,\ g_{2L}=b_Lb_L$.

Thus, the left hand side (LHS) of Equation (4.19) may be rewritten as:

$$F_0(z)F_1(z)[W_1(z^2)-W_0(z^2)]=[h_0(n_0-m_0)]+[h_1(n_0-m_0)]z^{-1}+\ldots+[h_{2L}(n_P-m_P)]z^{-2(L+P)}. \tag{4.29}$$

Next, using matrix notations, the coefficients for each z factor in the LHS of Equation (4.19) may be written as:

$$\underbrace{\begin{bmatrix} h_0 & 0 & \ldots & 0 \\ h_1 & 0 & & : \\ : & h_0 & & : \\ h_{2L} & : & & : \\ 0 & : & & : \\ : & h_{2L} & & 0 \\ : & 0 & & h_0 \\ : & : & & : \\ 0 & 0 & \ldots & h_{2L} \end{bmatrix}}_{[2(L+P)+1]\times[P+1]} . \underbrace{\begin{bmatrix} n_0 \\ : \\ n_P \end{bmatrix}}_{[(P+1)\times 1]} - \underbrace{\begin{bmatrix} h_0 & 0 & \ldots & 0 \\ h_1 & 0 & & : \\ : & h_0 & & : \\ h_{2L} & : & & : \\ 0 & : & & : \\ : & h_{2L} & & 0 \\ : & 0 & & h_0 \\ : & : & & : \\ 0 & 0 & \ldots & h_{2L} \end{bmatrix}}_{[2(L+P)+1]\times[P+1]} . \underbrace{\begin{bmatrix} m_0 \\ : \\ m_P \end{bmatrix}}_{[(P+1)\times 1]}, \tag{4.30}$$

which may be rewritten as:

$$\underset{[2(L+P)+1]\times[2(P+1)]}{\left[\begin{array}{cccc|cccc} h_0 & 0 & \dots & 0 & h_0 & 0 & \dots & 0 \\ h_1 & 0 & & \vdots & h_1 & 0 & & \vdots \\ \vdots & h_0 & & \vdots & \vdots & h_0 & & \vdots \\ h_{2L} & \vdots & & \vdots & h_{2L} & \vdots & & \vdots \\ 0 & \vdots & & \vdots & 0 & \vdots & & \vdots \\ \vdots & h_{2L} & & 0 & \vdots & h_{2L} & & 0 \\ \vdots & 0 & & h_0 & \vdots & 0 & & h_0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & h_{2L} & 0 & 0 & \dots & h_{2L} \end{array}\right]} . \underset{[2(P+1)\times1]}{\left[\begin{array}{c} n_0 \\ \vdots \\ n_P \\ \hline -m_0 \\ \vdots \\ -m_P \end{array}\right]} = \mathbf{B}. \tag{4.31}$$

Recall that the right hand side (RHS) of Equation (4.19) is expressed as:

$$F_1(z)^2K_1(z^2)\text{-}F_0(z)^2K_0(z^2). \tag{4.32}$$

Expanding $F_1(z)^2K_1(z^2)$ and $F_0(z)^2K_0(z^2)$ leads to:

$$F_1(z)^2K_1(z^2)=g_0z_0+(g_1z_0)z^{-1}+\dots+(g_{2L}z_P)z^{-2(L+P)}, \tag{4.33}$$

$$F_0(z)^2K_0(z^2)=k_0y_0+(k_1y_0)z^{-1}+\dots+(k_{2L}y_P)z^{-2(L+P)}. \tag{4.34}$$

Thus, the RHS of Equation (4.19) becomes:

$$F_1(z)^2K_1(z^2)\text{-}F_0(z)^2K_0(z^2)=(g_0z_0\text{-}k_0y_0)+(g_1z_0\text{-}k_1y_0)z^{-1}+\dots+(g_{2L}z_P\text{-}k_{2L}y_P)z^{-2(L+P)}. \tag{4.35}$$

In matrix notation, the coefficients for each z factor in Equation (4.35) can be written as:

$$\underset{[2(L+P)+1]\times[P+1]}{\begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & 0 & & \vdots \\ \vdots & g_0 & & \vdots \\ g_{2L} & \vdots & & \vdots \\ 0 & \vdots & & \vdots \\ \vdots & g_{2L} & & 0 \\ \vdots & 0 & & g_0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & g_{2L} \end{bmatrix}} . \underset{[(P+1)\times1]}{\begin{bmatrix} z_0 \\ \vdots \\ z_P \end{bmatrix}} - \underset{[2(L+P)+1]\times[P+1]}{\begin{bmatrix} k_0 & 0 & \dots & 0 \\ k_1 & 0 & & \vdots \\ \vdots & k_0 & & \vdots \\ k_{2L} & \vdots & & \vdots \\ 0 & \vdots & & \vdots \\ \vdots & k_{2L} & & 0 \\ \vdots & 0 & & k_0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & k_{2L} \end{bmatrix}} . \underset{[(P+1)\times1]}{\begin{bmatrix} y_0 \\ \vdots \\ y_P \end{bmatrix}}, \tag{4.36}$$

and,

$$\underbrace{\left[\begin{array}{cccc|cccc} g_0 & 0 & \cdots & 0 & k_0 & 0 & \cdots & 0 \\ g_1 & 0 & & \vdots & k_1 & 0 & & \vdots \\ \vdots & g_0 & & \vdots & \vdots & k_0 & & \vdots \\ g_{2L} & \vdots & & \vdots & k_{2L} & \vdots & & \vdots \\ 0 & \vdots & & \vdots & 0 & \vdots & & \vdots \\ \vdots & g_{2L} & & 0 & \vdots & k_{2L} & & 0 \\ \vdots & 0 & & g_0 & \vdots & 0 & & k_0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & g_{2L} & 0 & 0 & \cdots & k_{2L} \end{array}\right]}_{\substack{[2(L+P)+1]\times[2(P+1)] \\ \mathbf{A}}} \cdot \underbrace{\left[\begin{array}{c} z_0 \\ \vdots \\ z_P \\ \hline -y_0 \\ \vdots \\ -y_P \end{array}\right]}_{\substack{[2(P+1)\times 1] \\ \mathbf{x}}} . \tag{4.37}$$

Thus, Equation (4.19) can be expressed in matrix notation as:

$$\mathbf{Ax=B}. \tag{4.38}$$

Note that the matrix equation to be solved has a larger number of rows than columns. Therefore, the problem can only be solved in a least square sense. Thus, solving Equation (4.38) indicates that the aliasing will be minimized but not exactly canceled. Therefore, the least square solution is obtained by:

$$\mathbf{x=B\backslash A}, \tag{4.39}$$

where the vector **x** provides the coefficients for the filters $K_0(z)$ and $K_1(z)$.

Figure 4.11 illustrates the spectra of the noisy and the denoised signals obtained with the Wiener filter, the decimated Wavelet/Wiener (i.e., the scheme of section A without alias cancellation filters) and the decimated Wavelet/Wiener with alias cancellation analysis filters denoising schemes. The test signal is a sine wave with a length of 1024 samples at normalized frequency 0.2 and SNR level –3 dB. The length of the Wiener filter is 12 and the wavelet function is *sym8*. For the decimated Wavelet/Wiener denoising scheme of section A, a *1*-level decomposition is used, as well.

The plots show that aliases are canceled using the scheme with alias cancellation filters. As can be seen that the MSE value is also lower than that of the classic Wiener filter.

Figure 4.12 displays a portion of the time domain signals and Figure 4.13 shows the frequency responses of the band-specific Wiener and alias cancellation filters of the denoising scheme.

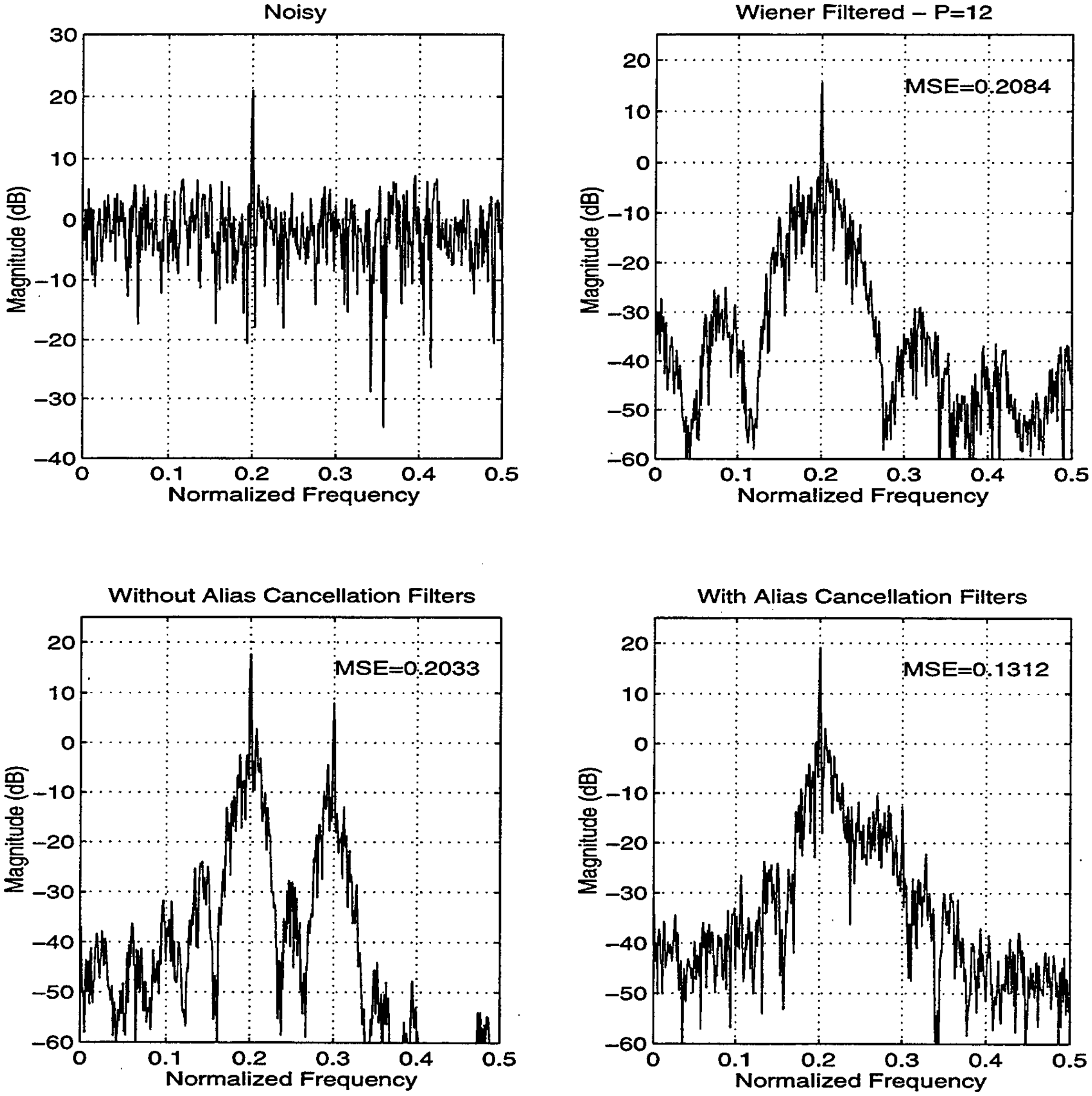


Figure 4.11: Algorithms comparison; One tone at frequency *f*=0.1, *sym8*.

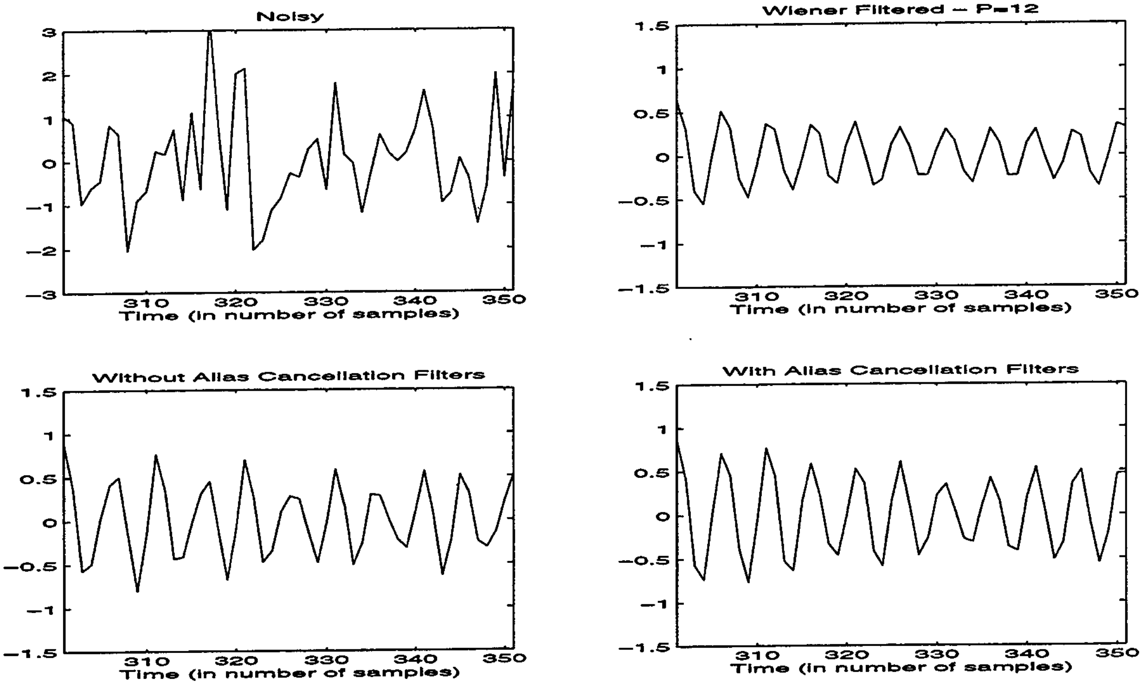


Figure 4.12: Time domain signals.

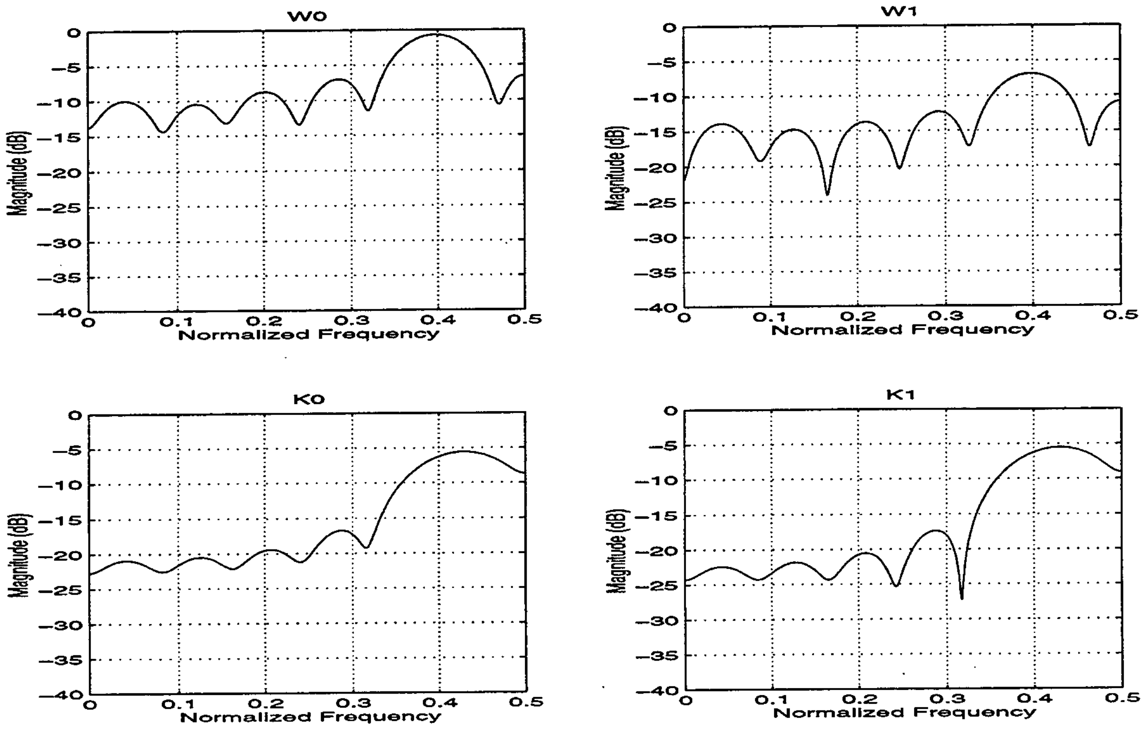


Figure 4.13: Wiener and alias cancellation filter characteristics.

Figure 4.14 shows that when using a *db10* wavelet function on the same signal as considered in Figure 4.11, the aliased term located at normalized frequency f=0.3 is not canceled. Our simulations showed that the amount of aliasing cancellation depends on the specific selection and combination of Wavelet and Wiener filter used.

Expansion of the scheme to multiple levels is left as a further study.

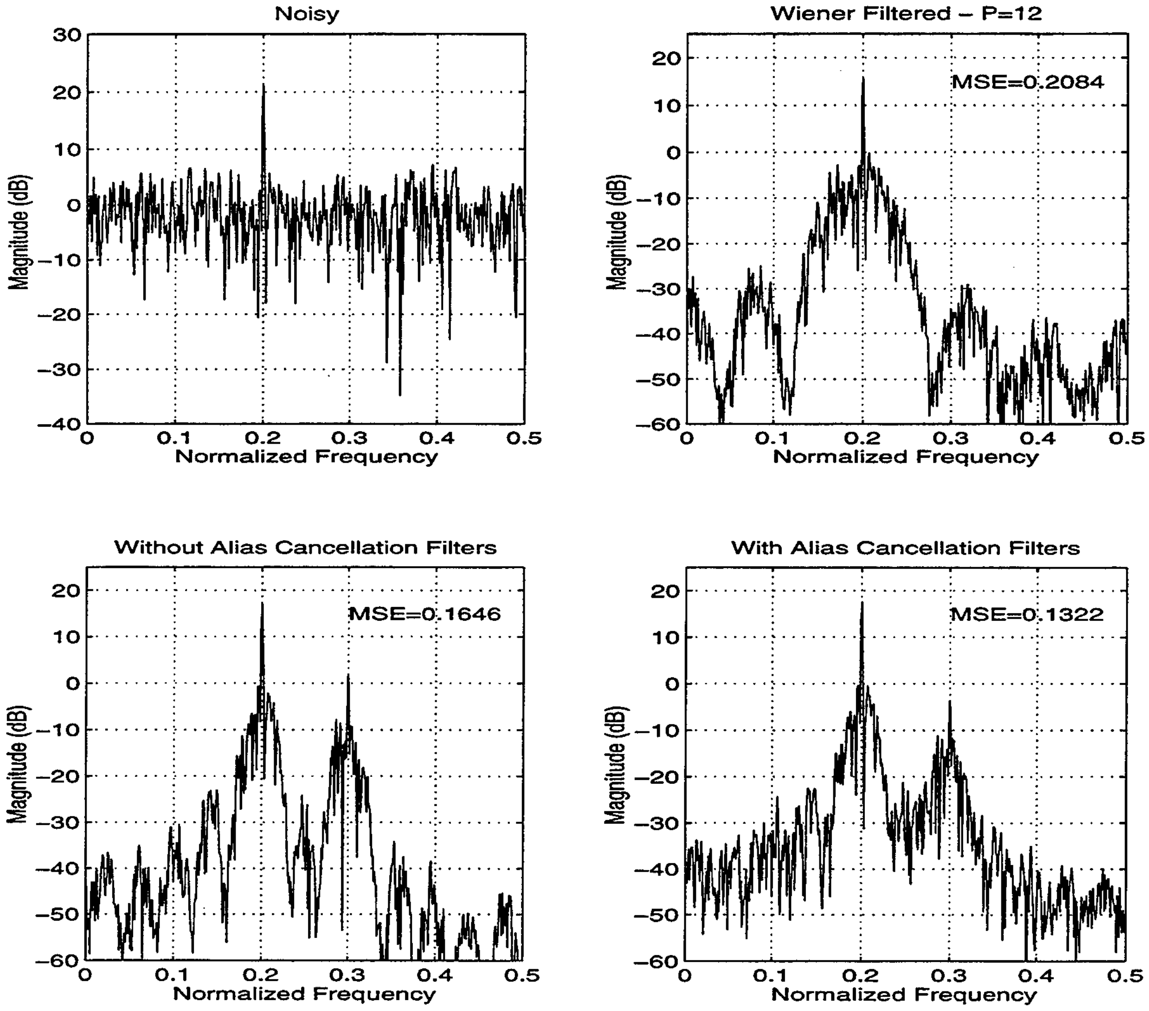


Figure 4.14: Algorithms comparison; One tone at frequency *f*=0.1, *db10.*

D. DENOISING USING NON-ORTHOGONAL UNDECIMATED WAVELET TRANSFORMS

This section considers the application of the non-orthogonal wavelet decomposition to the denoising of underwater signals. Recall that sampling the CWT in powers of two leads to the standard DWT (i.e., dyadic scales and positions) which is orthogonal, decimated and time-variant due to the decimation. Although those linear transforms are computationally efficient, design of analysis and synthesis filters is constrained by the orthogonality restriction, that is, the filter impulse responses are directly related to each other. However, non-orthogonal transforms have fewer restrictions on the decomposition filters, thus, better frequency resolution thus can be achieved than with orthogonal transforms.

The analyzing wavelet function used in the non-orthogonal transform is implemented using the undecimated A-trous algorithm and a modified version of the original Morlet wavelet [16,17] and given by:

$$\psi(t) = e^{j\eta t} e^{-\frac{\beta^2 t^2}{2}}, \qquad (4.40)$$

where η represents the center frequency of the mother wavelet and β represents the roll-off parameter, which is a constant proportional to the bandwidth of the analyzing wavelet. The Fourier transform of Equation (4.40) takes the following form:

$$\psi(\omega) = \frac{1}{\beta}\sqrt{2\pi} e^{-\frac{(\omega-\eta)^2}{2\beta^2}}. \qquad (4.41)$$

It is clear from Equation (4.40) that one can change the η and β parameters of the resulting decomposition, which results in different spectral partitioning.

The frequency resolution of the non-orthogonal transform may be modified by introducing subfilters (called *voices*) defined in each scale. These voices allow the user to vary the spectral partitioning of the transform [17]. Note that the partitioning of the orthogonal decompositions is fixed when a basis function is chosen. Thus, using non-orthogonal decomposition, much better narrowband frequency partitioning can be achieved than with orthogonal decompositions. However, a possible drawback with voices is the loss of temporal resolution due to the uncertainty principle, and an increase of computational cost per octave proportional to the number of voices selected. More details on analysis and synthesis operations of the non-orthogonal wavelet transform can be found in [17,18].

In this scheme, using the analyzing wavelet shown in Equation (4.40) will lead to complex DWT coefficients. Moreover, although those coefficients are associated with the white noise input, they are correlated unlike those of orthogonal transforms, which are uncorrelated.

Figure 4.15 display the spectral partitioning obtained by the A-trous decomposition with various roll-off parameters and number of voices for a 3-scale decomposition using a center frequency of 0.85π. Figure 4.16 depicts the 3-level spectral partitioning with different number of voices and center frequencies for a fixed roll-off factor of 0.15. In order to have a full coverage of the frequency axis, it is necessary to use a high roll-off parameter, which leads to larger overlap between the subfilters, as shown in Figure 4.15. Moreover, to cover very low frequencies one should use a high decomposition level. Finally, note that changing the center frequency η of the mother wavelet shifts the sub-filters in the spectrum, as shown in Figure 4.16.

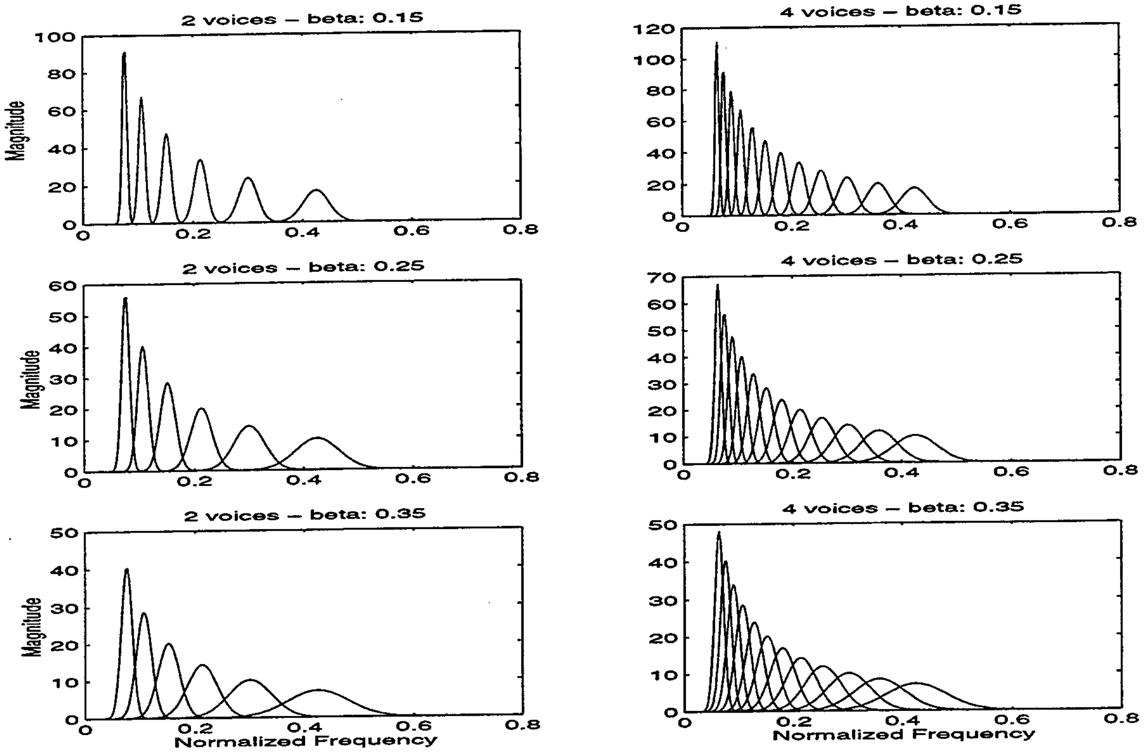


Figure 4.15: Spectral partitioning obtained for the A-trous algorithm, $\eta=0.85\pi$.

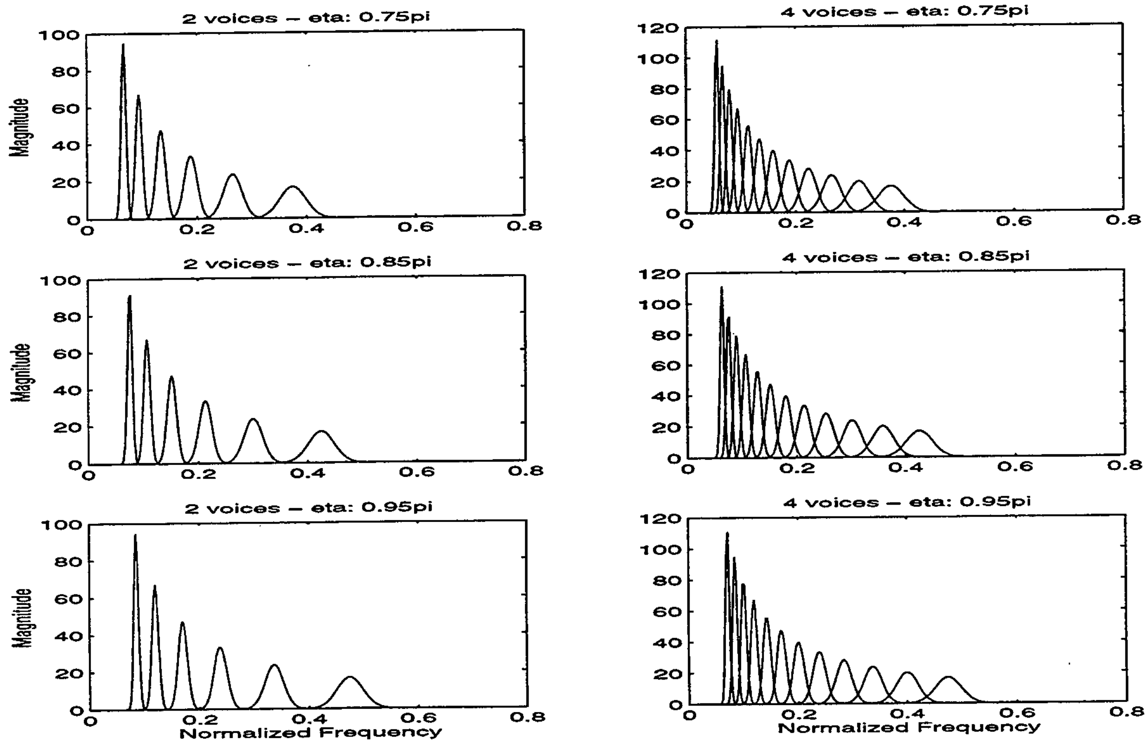


Figure 4.16: Spectral partitioning obtained for the A-trous algorithm, $\beta=0.15$.

The denoising performance of the non-orthogonal undecimated transform is illustrated in Figure 4.17. We applied a 5-scale decomposition to the noisy sinusoid, which is at normalized frequency 0.05 as shown earlier. The signal consists of 1024 samples and SNR level is –3 dB. We used 20 sub-filters in the spectrum (i.e., 4 voices per octave), a roll off factor 0.25 and a center frequency 0.75π. The resulting MSE is 0.1032. Recall that the classic Wiener filter gives a MSE of 0.1994 for a filter length of 12.

Simulations showed that this implementation introduces amplitude distortion like the scheme investigated in section B. Thus, we normalized the signals to unit energy for ease of comparisons.

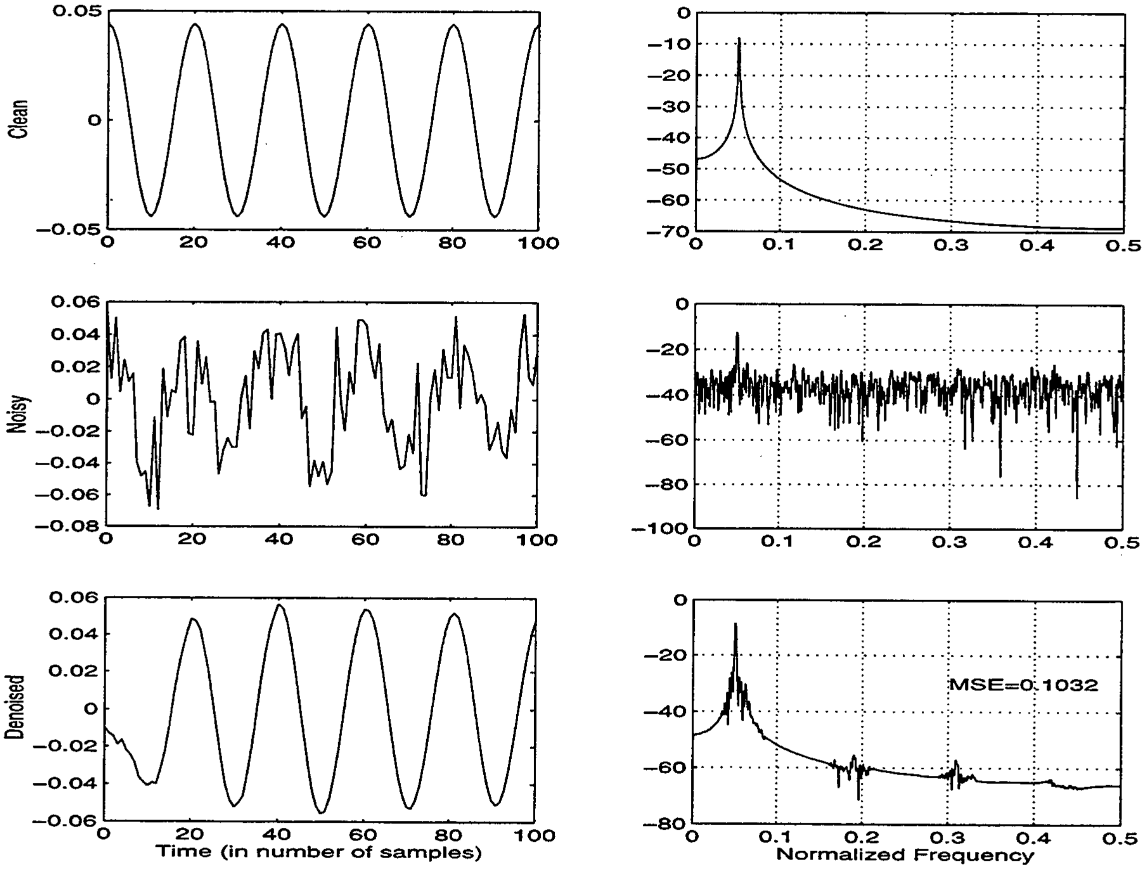


Figure 4.17: Denoising with non-orthogonal undecimated wavelet transform.

E. METHOD COMPARISONS

In this section, two synthetic signals, *Doppler* and *Decay*, are studied to compare the different methods. These signals are of interest to us since they have some essential features of real world acoustic signals. The signals, consist of 1024 data points and have an SNR of –3 dB, as shown in Figure 4.18 and Figure 4.19, respectively.

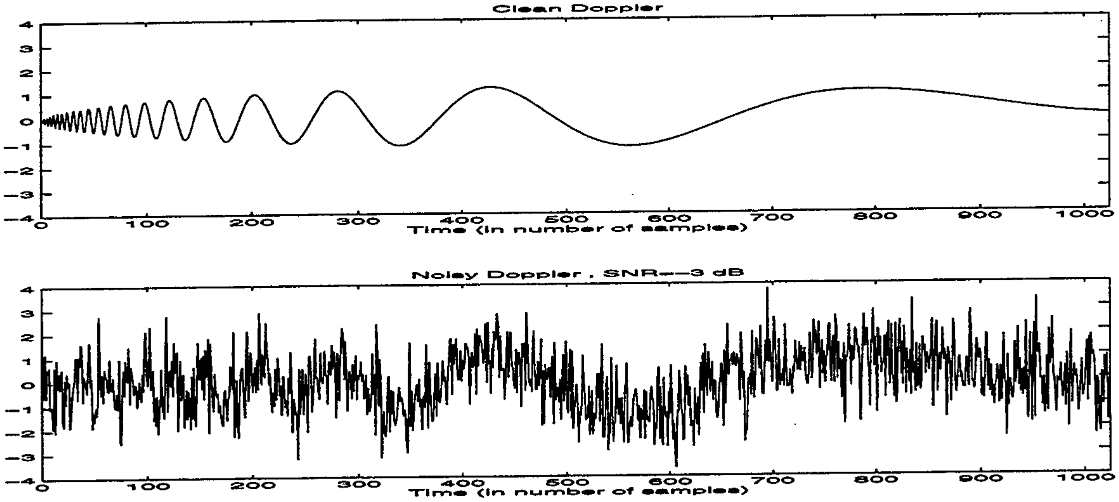


Figure 4.18: The clean and noisy Doppler, SNR=-3 dB.

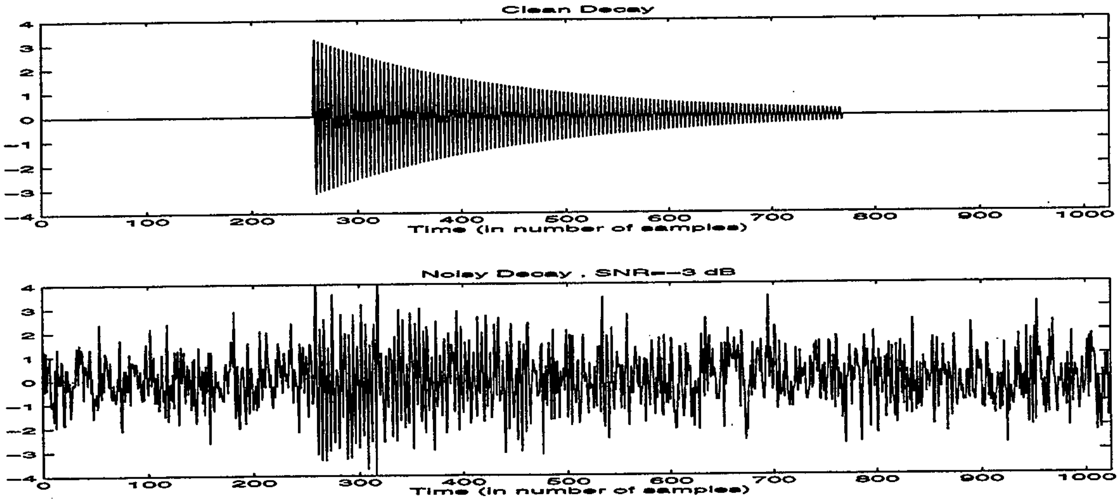


Figure 4.19: The clean and noisy Decay, SNR=-3 dB.

The denoising schemes compared and parameters used for *Doppler* are:

1- Wiener filter with a length of 12,

2- Wavelet thresholding with *5*-level *sym8* decomposition using soft heursure thresholding method,

3- Decimated Wavelet/Wiener filtering with *5*-level *sym8* decomposition using a multiresolution Wiener filter length of 12,

4- Undecimated Wavelet/Wiener filtering with *1*-level *sym8* decomposition using a band-specific Wiener filter length of 12,

5- Decimated Wavelet/Wiener filtering with alias cancellation filters (i.e., $K_k(z)$ where k=0,1) and *1*-level *db10* (since it performed better than *sym8* for the *Doppler* test signal) decomposition using a band-specific Wiener filter length of 12,

6- Non-orthogonal undecimated wavelet transform with *5*-level decomposition and a Wiener filter length of 12, 2 voices per octave using a roll-off factor β of 0.35 and a center frequency η of 0.85π.

The resulting denoised signals and MSEs obtained for the energy normalized signals are displayed with the order shown above in Figure 4.20. Note that the undecimated scheme leads to amplitude distortion in the denoised signal as expected. On the other hand, the A-trous algorithm does not have an amplitude distortion but a delay unlike the situation in the previous section. Moreover, it has a worse MSE value than any other schemes due to the delay. As a general comment, it can be said that all denoising schemes lead to similar results except the A-trous algorithm. In addition, it is clear from Figure 4.20 that there is difference between visual and numerical quality of denoising.

Although the A-trous, thresholding and decimated Wavelet/Wiener denoising schemes achieve the highest MSEs, they provide a smoother visual reproduction of the clean signal.

Next, we applied the denoising schemes mentioned earlier to the second test signal *Decay*. For the wavelet thresholding scheme, different wavelet filters were tried. Since *db10* gave better results, it was used instead of *sym8*. In addition, a roll-off factor of 0.25 and a center frequency of 0.95π were used for the non-orthogonal undecimated wavelet denoising. All remaining parameters are the same as those of the *Doppler* signal experiment.

Figure 4.21 depicts the denoised signals and the corresponding MSEs. As expected, the undecimated schemes have amplitude distortion. In addition, wavelet thresholding significantly performed worse than the other schemes.

Note that knowledge of the exact noise-only segment was assumed in the simulations conducted above as mentioned in Chapter III, section C. Results showed that performance degradations occurred in the denoising schemes in absence of this assumption, as expected [15].

Finally, we note that one can make performance improvements by choosing parameters based on experimental evidence.

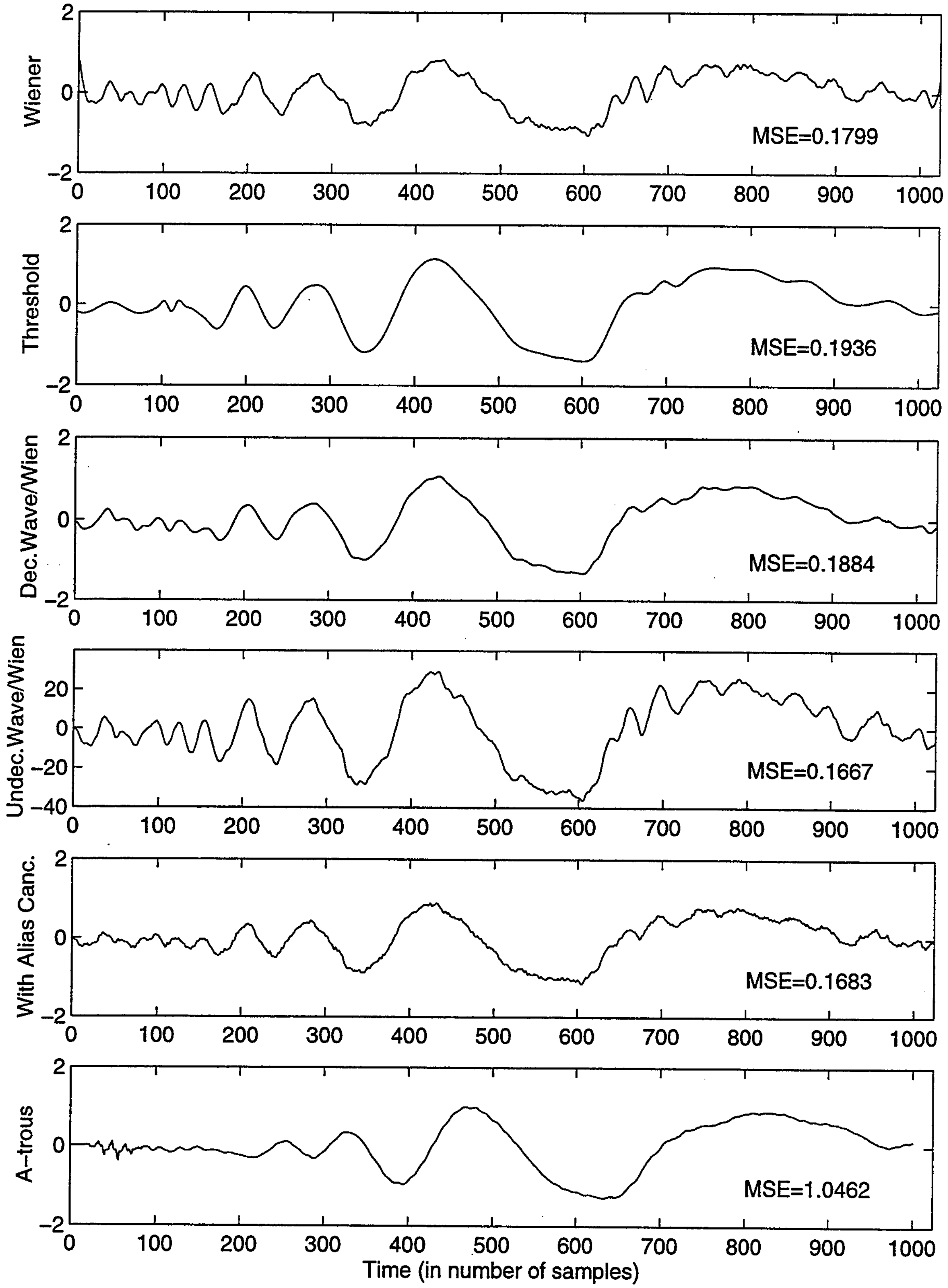


Figure 4.20: Performance comparisons of the denoising algorithms. Note the disparity between MSE and visual quality.

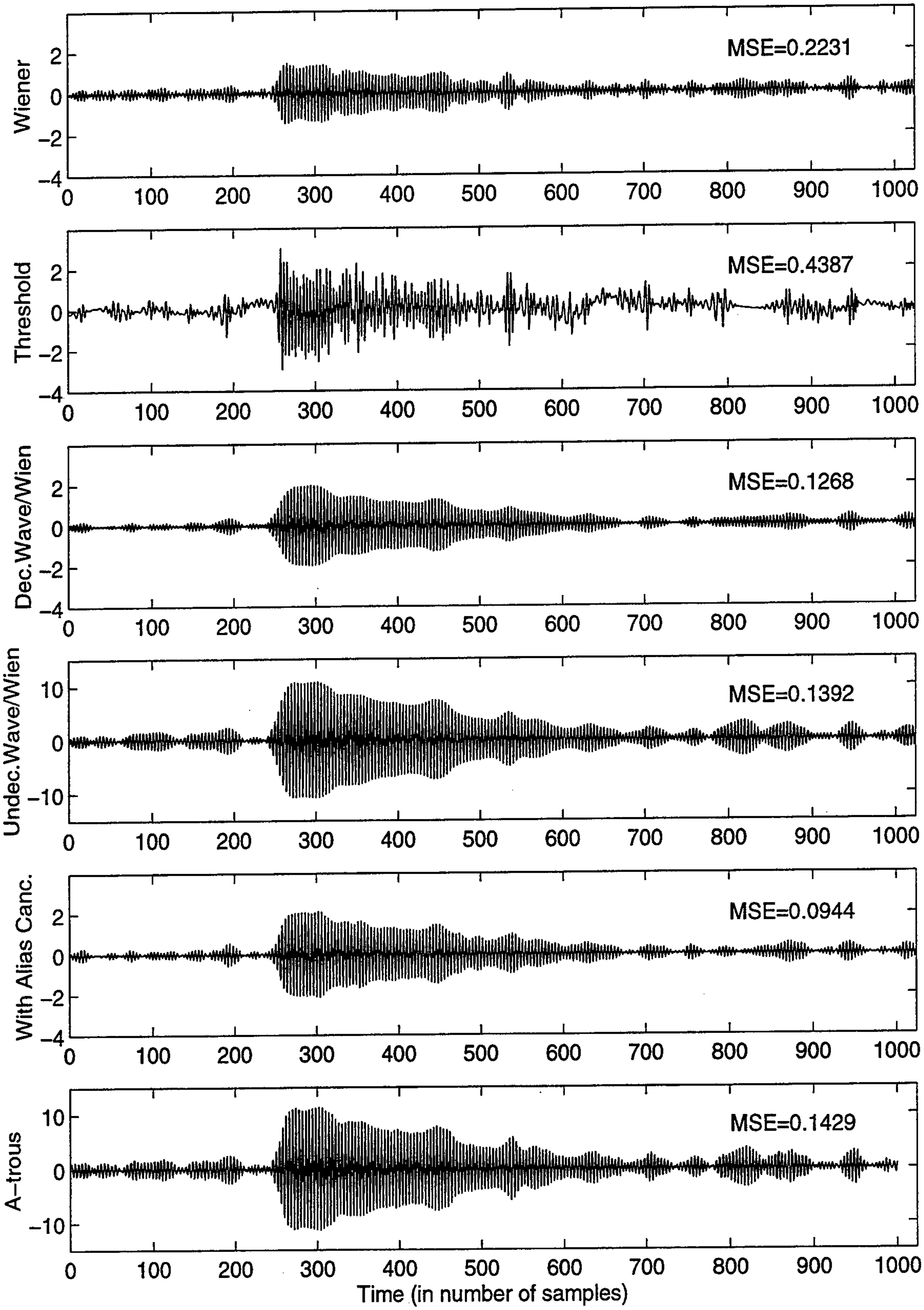


Figure 4.21: Performance comparisons of the denoising algorithms.

V. CONCLUSIONS

In this thesis, we investigated a number of denoising schemes by combining wavelets and Wiener filtering techniques. The performance comparison of those schemes to that of the classic Wiener filter and wavelet thresholding was conducted by applying them to the problem of acoustic noise removal.

In general, simulations show that the Wiener filter with an appropriate length outperforms other denoising techniques in terms of MSE, especially at lower SNR levels.

The combined Wavelet and Wiener filter denoising scheme also performs well under more restrictive conditions. However, the performance is highly dependent on signal of interest, accurate estimation of noise and wavelet basis function used. Aliasing is another problem with this type of schemes due to the introduction of Wiener filters, although they provide low MSEs.

Two types of alias cancellation filters were conducted: single-band post synthesis and cross-band filters. Results showed that the single-band post filters reduce the aliasing component but introduce some additional distortion in the denoised signal. Cross filter transfer functions were obtained as the result of solving a least square matrix equation which showed that this option leads to minimization of the aliased components but not to their exact cancellation. Results showed this scheme performance to be quite sensitive to the choice of wavelet filter type, Wiener filter order and signal frequency location. Thus, considering the added problem generated by the alias canceling filters, users should consider using the denoising schemes without any added alias canceling filters when some amount of aliasing is permissible.

Finally, the undecimated A-trous Wavelet/Wiener denoising scheme was considered. Simulations showed better performance results than those obtained with the classical Wiener filter, however, the computational cost is much higher.

LIST OF REFERENCES

[1] R. Barsanti, *Denoising of Ocean Acoustic Signals Using Wavelet-Based Techniques*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1996.

[2] C. Therrien, *Discrete Random Signals and Statistical Signal Processing,* Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[3] A. Bruce, D. Donoho, and H. Gao, "Wavelet Analysis," *IEEE Spectrum*, pp. 26-35, October 1996.

[4] D. Gabor, "Theory of Communication," *J. of the IEE, Vol.93*, pp. 429-457, 1946.

[5] S. Haykin, *Communication Systems*, John Wiley & Sons, New York, 1994.

[6] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, pp. 14-38, October 1991.

[7] *The Matlab Wavelet Toolbox*, The Mathworks, Inc., Natick, MA, 1996.

[8] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pat. Anal. Mach. Intell*, Vol. 11, pp. 674-693, 1989.

[9] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, 1996.

[10] M. P. Fargues, "Statistical Digital Signal Processing" class notes from EC 3420, Naval Postgraduate School, Monterey, CA, Winter 1997.

[11] K. Frack, *Improving Transient Signal Synthesis Through Noise Modeling and Noise Removal*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1994.

[12] D. Donoho and I. Jonnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, Vol. 81, pp. 425-455, 1994.

[13] D. Donoho, "De-noising by soft thresholding," *IEEE Trans. Inform. Theory*, Vol. 41, pp. 613-627, May 1995.

[14] D. Donoho and I. Jonnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Stat. Assoc.*, Vol. 90, pp. 1200-1224, Dec. 1995.

[15] F. Forney, *Acoustic Noise Removal by Combining Wiener and Wavelet Filtering Techniques*, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1998.

[16] W. Brooks, *Ultra-Wideband Radar Transient Detection Using Time-Frequency And Wavelet Transforms*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1992.

[17] M. Shensa, “The Discrete Wavelet Transform: Wedding the A-trous and Mallat Algorithms,” *IEEE Trans. on Signal Processing*, Vol. 40, pp. 2464-2482, October 1992.

[18] M. Shensa, “Discrete Inverses for Nonorthogonal Transforms,” *IEEE Trans. on Signal Processing*, Vol. 44, Apr. 1996.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center..2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library..2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC...1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Prof. Monique P. Fargues, Code EC/Fa...2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Prof. Ralph Hippenstiel, Code EC/Hi..1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Deniz Kuvvetleri Komutanligi...2
   Personel Daire Baskanligi
   Bakanliklar
   Ankara, TURKEY

7. Deniz Harp Okulu Komutanligi..1
   Kutuphane
   Tuzla, Istanbul- 81704, TURKEY

8. Coskun Cebeci...2
   Kozyatagi 19 Mayis Mah. Yildiz Sok.
   Yildiz Apt. No:16 D.45
   Kadikoy,Istanbul-81090, TURKEY